Tampere University

Mario Gómez

COMPARATIVE ANALYSIS OF NEURONAL SEGMENTATION METHODS FOR SINGLE CELL SIGNAL EXTRACTION

Master Thesis

Faculty of Medicine and Health Technology
Master of Biomedical Sciences and Engineering
June 2019

ABSTRACT

Mario Gómez: Comparative Analysis of Neuronal Segmentation Methods for Single Cell Signal Extraction
Master of Biomedical Sciences and Engineering
Tampere University
Master Degree
June 2019

In the Molecular Signaling Laboratory (MSLab), when working with neuronal cells that have been treated with a dye agent, a parameter extraction protocol is followed, mainly the intensity of the image, which requires advanced knowledge in programming languages and tools, as well as a prudent time to extract the information. The investigator, on most occasions, is limited by its researcher background.

In this work, the master degree student has developed a tool that offers the extraction of the results, without necessitating the knowledge in image processing languages, and exposes them in plots that make it easier the interpretation for the investigator. This software also allows the export of the results in an Excel file.

On this project, a method has been implemented that performs cellular segmentation and extracts the information in an image processing language, and desktop software that uses that method, transparently to the researcher, and exposes the results in graphs.

Keywords: signal, processing, segmentation, images, method, biomedical, glioma, neuronal, cancer, cancerous, cells, thesis.

PREFACE

The basis of this work has been developed by my eagerness in providing solutions to problems raised in the field of Biomedicine, especially in research on cancer-related illnesses, starting with the final thesis project of my bachelor degree and continuing with the final thesis project of my master degree.

All my thanks to the team of the Molecular Signaling Laboratory (MSLab) of the Faculty of Medicine and Health Technology of Tampere University, for all the knowledge and experience offered. In particular, to the professor and group leader, Meenakshisundaram Kandhavelu, who has been able to drive, in the best possible way, my passion for working and contributing in the fight against cancer.

Tampere, 3rd June 2019

Mario Gómez

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

1321N1	Cell line from human brain.
Backend	Part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data.
CellProfiler	Software open source designed to enable biologist, without training in image processing languages, quantitatively measure.
CMSM	Comparative-Mask Segmentation Method.
C#	Software open source designed to enable biologist, without training in image processing languages, quantitatively measure.
Frontend	Part of a computer system or application with which the user interacts directly.
GFPmut3	GFPmut3 is a fluorescent protein published in 1996, derived from Aequorea victoria.
IDE	Integrated Development Environment.
MATLAB	Matrix Laboratory. Programming language used in image processing.
MO filter	Morphological Opening Filter.
MSLab	Molecular Signaling Laboratory (Faculty of Medicine and Health Technology of Tampere University).
PCs	Personal Computers.
UGR	Univeristy of Granada (Spain).
UI	User Interface.
UNIZAR	University of Zaragoza (Spain).
UPNA	Public University of Navarra (Spain).
URL	Uniform Resource Locator

1 INTRODUCTION

This thesis is a work that collects the problems of a group of researchers from MSLab when they study and extract information from images while they are working with images of neuronal cells, treated with a contrast agent. Besides, it presents a viable solution that presents advantages and solves human aspects to increase the precision and reliability with which conclusions are extracted from the results of the study.

1.1 Background of the Master Degree Student

As the author of this work and **graduated in Telecommunications Engineering** by *University of Granada (UGR)*, in Spain, I had the first contact with biomedicine in my Bachelors Degree Thesis *"Design of a medical device for magnetic induction for the treatment of cancer"* and I begin to delve into the field.

My interest in the matter and my enthusiasm for providing solutions to the problems that arise provoke I study a **Master Degree in Biomedical Engineering** at *Public University of Navarra (UPNA)*, carrying out the **specialization in "Biomechanics and Biomaterials"** through the *University of Zaragoza (UNIZAR)*.

Master Degree Student Technical Skills

- Signal processing.
- Image analysis.
- Tumor diseases.
- Tumor tissues.
- IT programming.

Figure 1.1. *Skills of the Master Degree Thesis author.*

1.2 Context and Problems

In the MATLAB, a specific protocol is used by the researchers when they work with neuronal cells images, treated with a dye agent, and whose relevant information want to be extracted. This protocol has certain limitations when it is implemented since it combines image processing techniques that considerably difficult that task to the researchers, whose professional profile (biologist, chemist, pharmacology or others) is not related.

The process of extracting information is limited by the intensity of the image, using a programming language (MATLAB) widely used in the field of image processing research. The count of cells is done manually by the researcher, counting one by one. When it wants to extract information about the size, it has to make an estimation using the approximate dimensions of the cell and the image scale. The size requires a considerable time that, on many occasions, the researcher does not have.

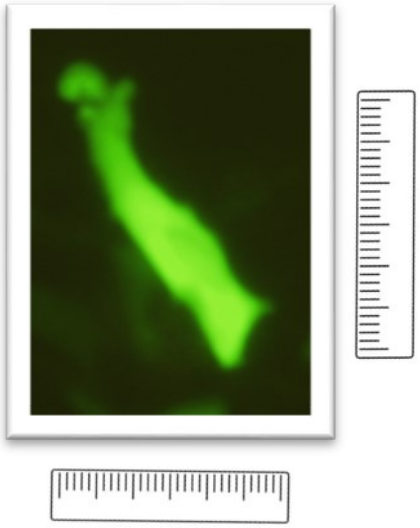


Figure 1.2. *Manual process to get the cell size.*

MSLab - Problems and Limitations

- The process requires knowledge on image processing.
- The information extracted from images are restricted to intensity (and size in a few cases).
- To get the results is necessary the intervention of the researcher.
- The process is manual. It introduces human errors.
- It gets long time if the number of cells are high.
- Sometimes, the contrast of images are very poor.
- The use of MATLAB is restricted to some countries and the license is really expensive.
- Once the information is gathered, the file format with the information depends on the researcher.
- The information has to be gathered image by image.

Figure 1.3. *Limitations and problems working in MSLab with neuronal images and contrast.*

1.3 The Solution to the Problem

In order to solve all problems, it is proposed in this work the implementation of software, installable in any PC of the laboratory, and whose management requires the minimum knowledge by the investigator.

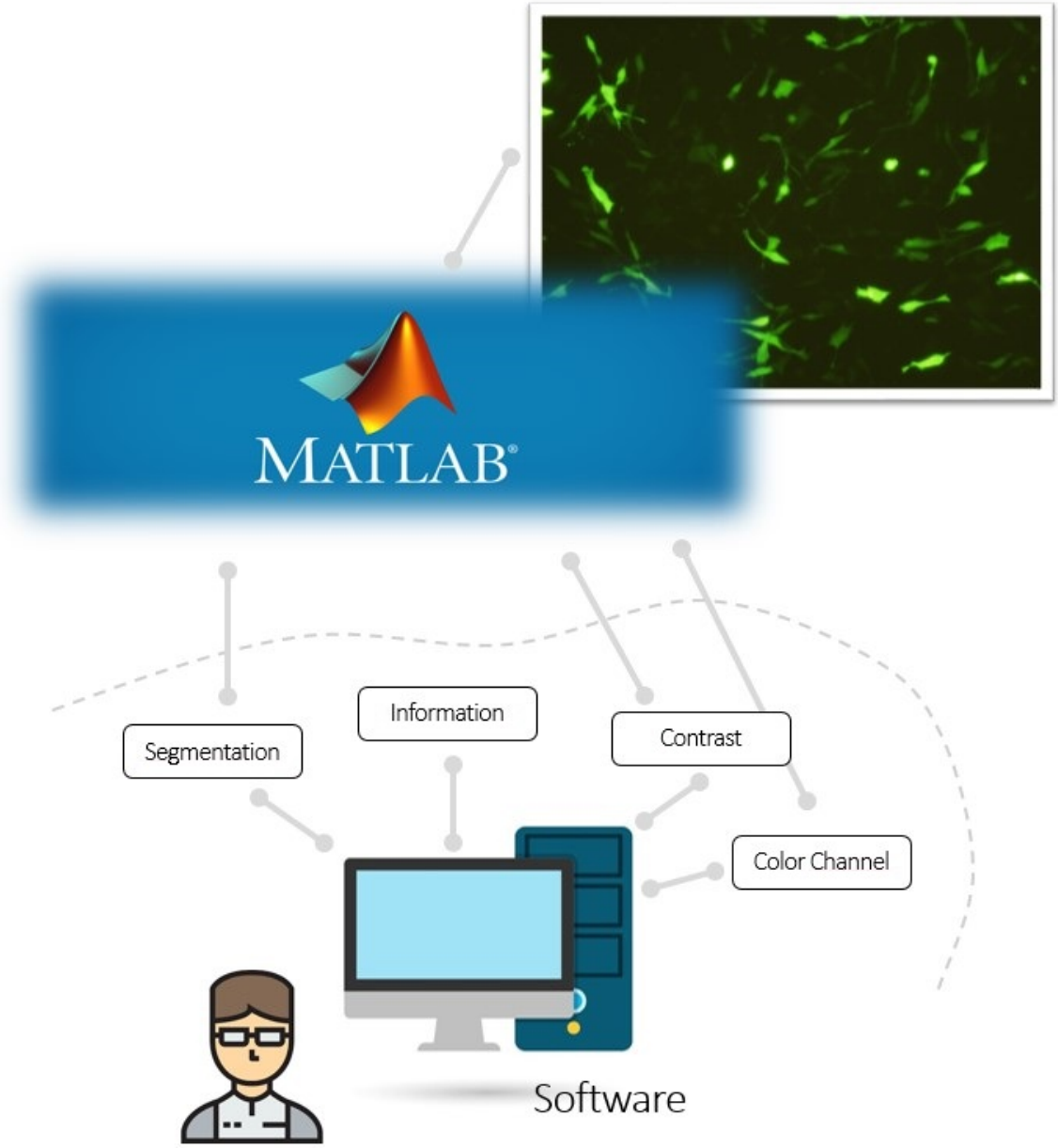
The capacity of the software is no longer limited to extraction of intensity but extends to others like orientation, the neuron eccentricity or its position. As for the area of the cell, the same method is used for its calculation, eliminating the intervention of the investigator in each case and obtaining a more precise measure of the real value.

In addition to the process of segmenting the image, which the information is extracted later, other modules have been added to improve the quality of the image. Filters have been introduced to extract different channels from the image. This is useful when working with images of cells that have been treated with several contrast dyes since each one represents a physical magnitude or a different cellular process.

Solution to the problem

- **Implementation of a software application:** The application uses an easy UI running MATLAB in background. It lets the researcher extract the information quickly and eliminates the need to know about image processing.
- **Software modules:** Different modules are required for segmentation, for information extraction, for information representation, for contrast improving, for color channel isolation, for exportation into a file.
- **Segmentation module:** The segmentation is adapted to neuronal cells characteristics.
- **Information module:** It provides relevant information (area, orientation, eccentricity, intensity value of each cell and mean parameters of the cell set in each image, in a easy way to evaluate the results and lets the research export that information into a file.
- **Contrast module:** It improve the contrast of the images. Sometimes, cell intensity is very poor, however, it lets the researcher work with this kind of images.
- **Color channel isolation module:** It lets gather intensity information of neuronal cells treated with different dyes, and different cell processes or parameters can be obtained without interference.

(a) *Module description of the solution.*



(b) *Diagram of solution.*

Figure 1.4. *The solution to the problems of section 1.2.*

2 SEGMENTATION

To implement the segmentation of neuronal cells is necessary, first of all, to know the characteristics of the images with which we are going to treat. Secondly, to choose the software that best suits the needs to, finally, implement a method capable of detecting and segmenting the neurons and able to extract relevant information.

2.1 Cell Types and Input Images

The cell line we're going to describe here is **1321N1**. The dye that is displayed in this type of images is the protein **GFPmut3** that offers images of green colour mainly.

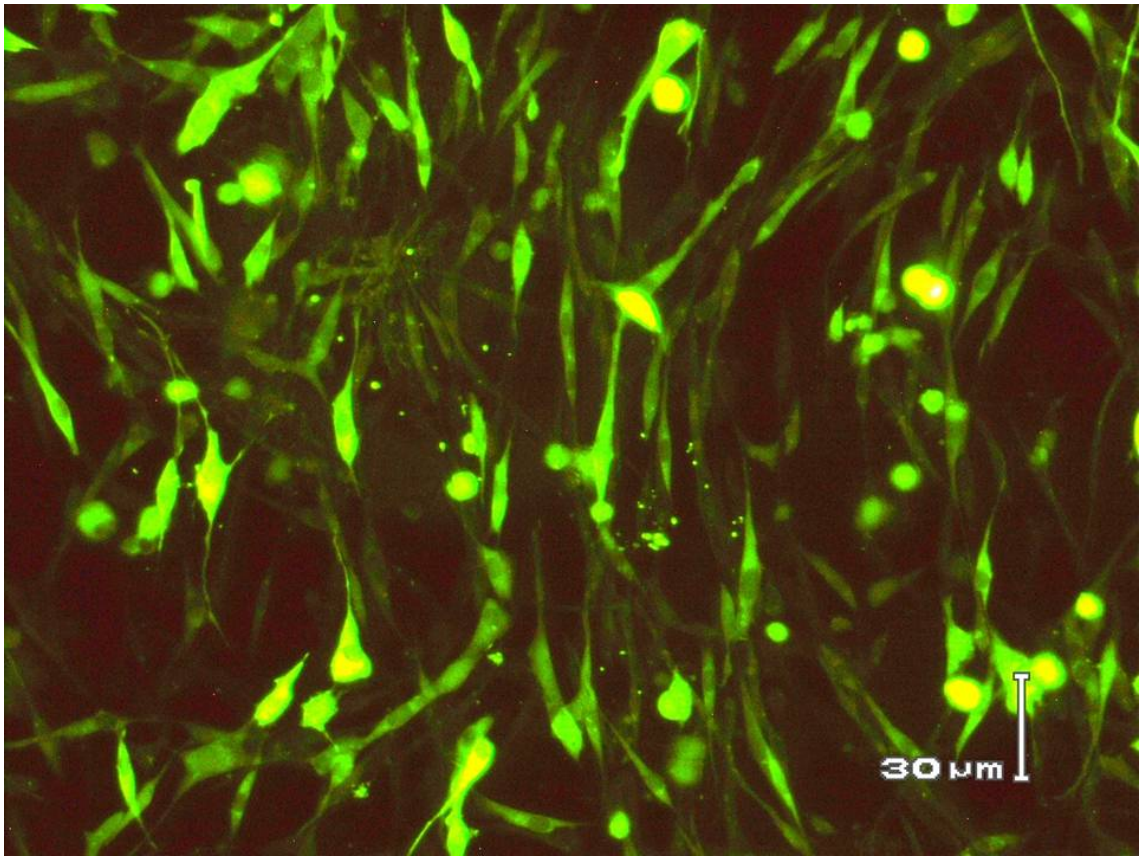


Figure 2.1. *1321N1 cell line from human brain.*

Image Characteristics

- Green channel images.
- Black background.
- Solid body.
- Good definitions borders.
- High contrast between cells and background.
- The file extension usually is TIFF format.

Figure 2.2. *1321N1 treated with GFPmut3 Image Characteristics.*

2.2 Image Processing Software Discussion

In MSLab, researchers have been using open source software to extract information from images, **CellProfiler**. This software was implemented as a useful tool for research in the field of biology requiring computer-aided vision or quantitative measurement on multiple images.

However, being a general-purpose software, the number of configurable parameters is so extensive that it requires prior training by the researchers who will use it. Besides, the cellular segmentation that implements this software is quite basic, being quite low the number of cells detected. This software does not have a module implemented to represent the obtained data. They are directly exported to a CSV file (see Figure 2.3).

To address all the needs of the laboratory, it has been decided to implement new software that is easier to use, that improves the segmentation when dealing with this type of cells and that allows an instantaneous visualization of the results obtained.

The idea of improving CellProfiler has been ruled out because addressing the transformation requires excessive time (in addition to the implementation of the new modules requires a considerable amount of time to understand the operation of the code already implemented).

Since we are going to implement the application from the ground up, it has been decided to use a programming language that allows the encapsulation of the code that will do the image treatment, and that would act behind a desktop application (see Figure 1.4). Because of its versatility and the vast number of functions to work with images, **MATLAB** has been decided to use for image segmentation.

The desktop program, which will act as a straightforward user interface, between the MATLAB code and the researcher, has been decided to be done in the **C#** language because all the modern machines (and planned in the future) use Windows as the operating system. Using this native Windows language allows the smooth integration of MATLAB into our desktop application (see Figure 2.4).

Debilities of CellProfiler

- General purpose (for biologist) software.
- The segmentation of this kind of cells are poor.
- High number of configurable parameters.
- It does not have a data visualization module.
- It requires training to be used by researchers.

Figure 2.3. *Weakness of the software used in MSLab, CellProfiler.*

Opportunities of the new software

- Specific software for MSLab tasks.
- Segmentation improving for MSLab cell line.
- The number of configurable parameters are very reduced.
- Addition of a data visualization module.
- The software training for researchers is reduced to minimum.

Figure 2.4. *Opportunities of the new implementation software.*

2.3 Other Segmentation Methods (previously implemented)

The first step is to collect information about other methods that are used for cell segmentation. In [2] *Chowdhury et al. BMC Bioinformatics 2013*, the process begins with a filter that eliminates image noise (we call image noise to everything that is not information such as dust spots, poorly defined pixels, or intensity artifacts that reduce quality). Then a first segmentation is applied to highlight the information and discard the background. On this information, a second segmentation (*"Multi-scale morphological edge detection"*) is made, which is the one that segments the cell and gives it form by applying the necessary corrections (see Figure 2.5).

2.4 Comparative-Mask Segmentation Method (CMSM)

To implement our method, we implement a process quite similar to used in Figure 2.5 (see Figure 2.6). We start by applying a filter to the image to treat it whatever the intensity channel. This makes the application independent of the dye agent used in the experiment. Then, we apply to the image a Wiener filter to reduce the noise. The morphological opening increases the contrast between the black background and the information (cells). Before the initial segmentation is performed, the image is quantized. That exceeds the threshold set by the minimum threshold parameter will depend on whether it is consid-

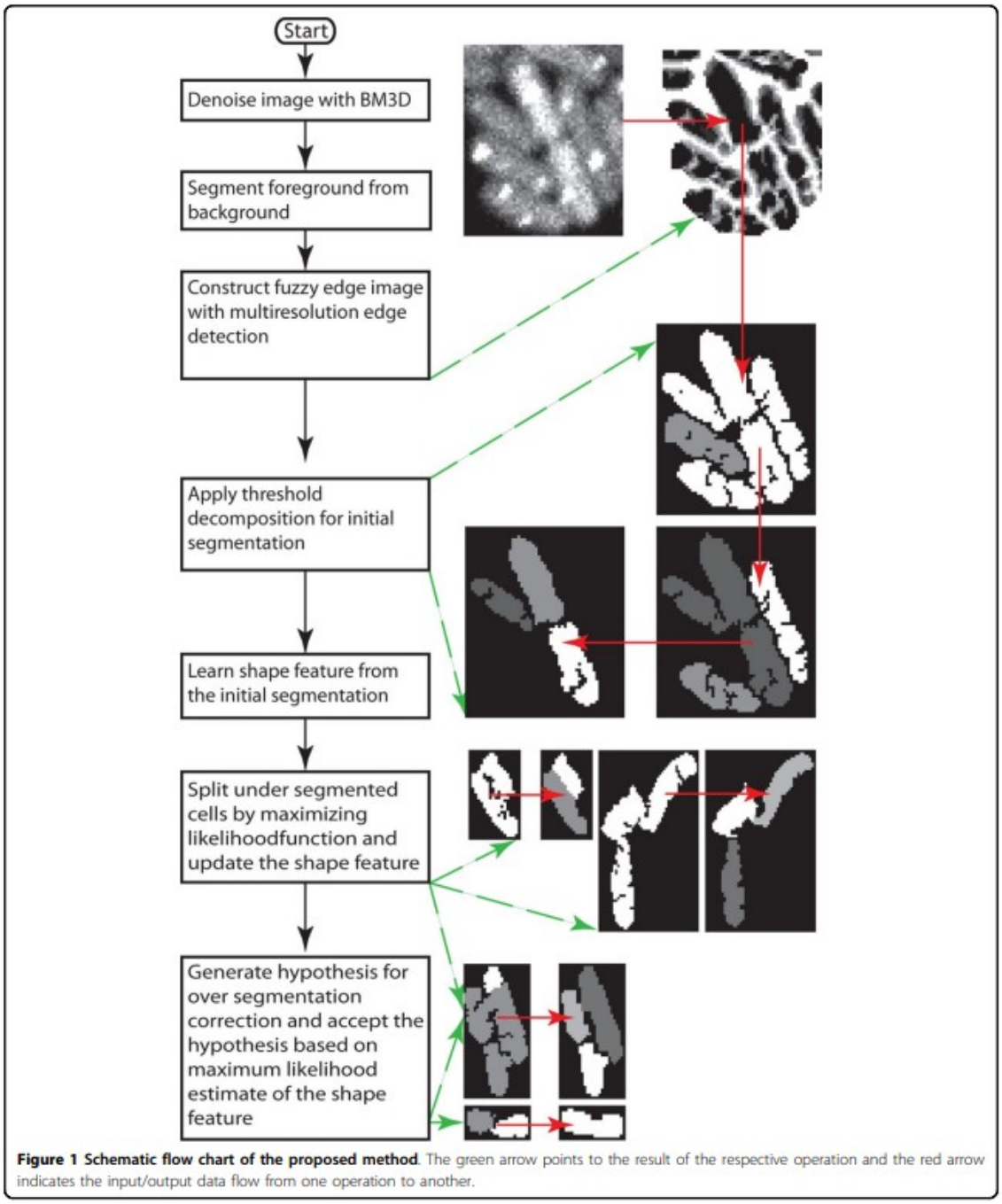
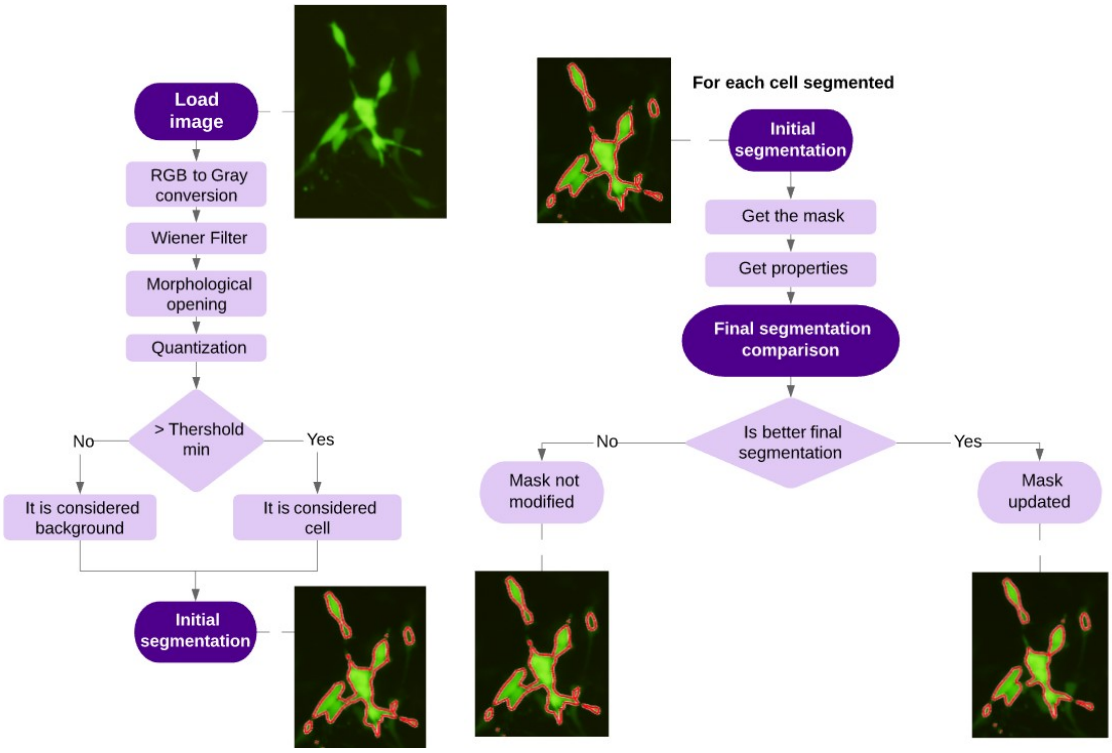


Figure 1 Schematic flow chart of the proposed method. The green arrow points to the result of the respective operation and the red arrow indicates the input/output data flow from one operation to another.

Figure 2.5. *Schematic flow chart of the MAMLE method from the article S. Chowdhury, M. Kandhavelu, O. Yli-Harja and A. S. Ribeiro. Cell segmentation by multi-resolution analysis and maximum likelihood estimation (MAMLE).BMC Bioin-formatics 2013.*

ered as background or as information. After initial segmentation, many segmented cells are obtained. For each of them, we obtain the mask, we obtain the properties of the Mask, and we apply the second segmentation called **"Comparative-Mask Segmentation Method (CMSM)"**. If the final segmentation has modified the mask, it is updated if the user chooses it (because the final segmentation improves the first one) or the mask is retained from the first segment otherwise.



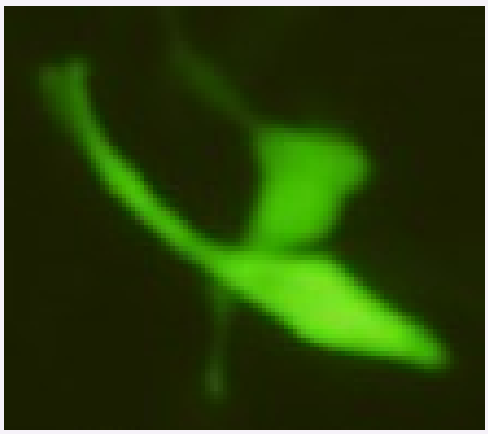
(a) *Flux diagram of the initial segmentation (first step).* ***(b)*** *Flux diagram of the second segmentation (final step).*

Figure 2.6. *Complete flux diagram of Comparative-Mask Segmentation Method.*

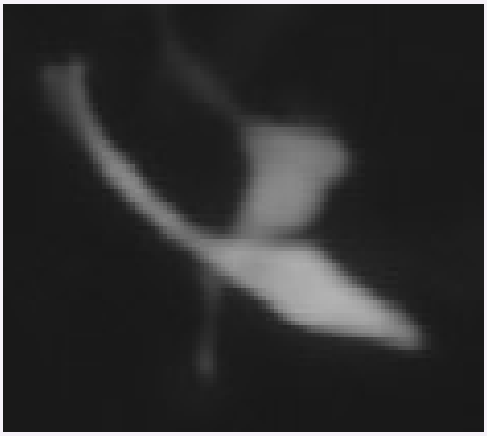
Note: For the implementation of the method, information has been collected in different image segmentation articles, both for CellProfiler *[1] Carpenter et al.* and MATLAB *[3] Hodneland et al.*

CMSM Process

- **From RGB to Gray fiter:** We convert the three-channel image (red, green, blue) into a single-channel image (grayscale). With this, we make sure that our method is independent of the dye agent used in the experiment. This is very useful because it reduces the processing time to 33% of the time it would take for each of the channels in the image.

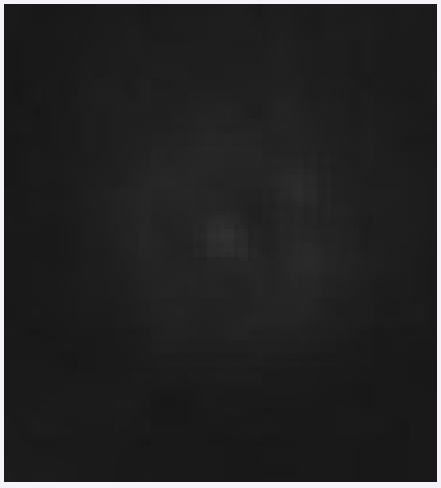


RGB Image.



Gray scale Image.

- **Wienner filter:** This filter minimizes the half-square error between the expected value and the actual value of a pixel in the image. That is, it minimizes the noise produced by specks and those loose pixels whose value differs significantly from the expected value of its environment.

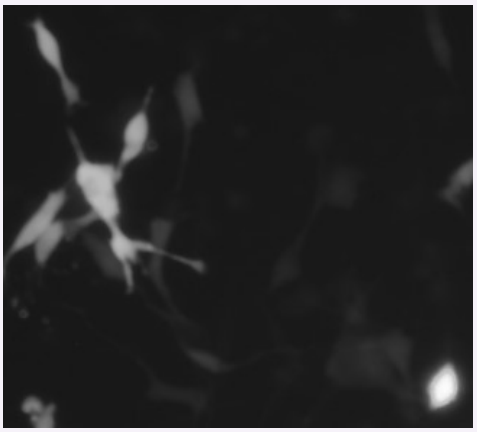


Before Wienner filter.

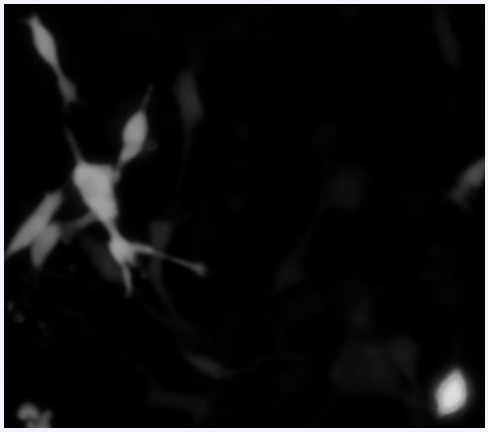
After Wienner filter.

- **Morphological Opening:** This process basically causes a contrast increment between what is considered background (low light intensity) and what is considered information (high cell intensity).

CMSM Process (cont.)



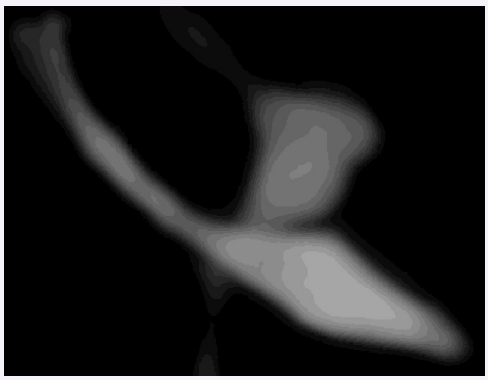
Before MO filter.



After MO filter.

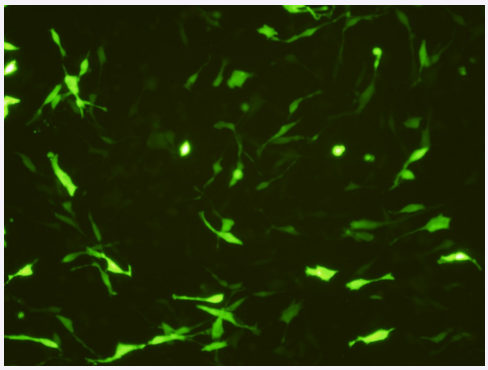
- **Image Quantization:** This process reduces the image information from 255 levels of intensity to a reasonably small number. These levels maintain the morphology of the cell, that is, the lower levels of intensity could be considered the membranous wall of the cell, the intermediate layers the cytoplasm, and the highest the nucleus.

Before Quantization.

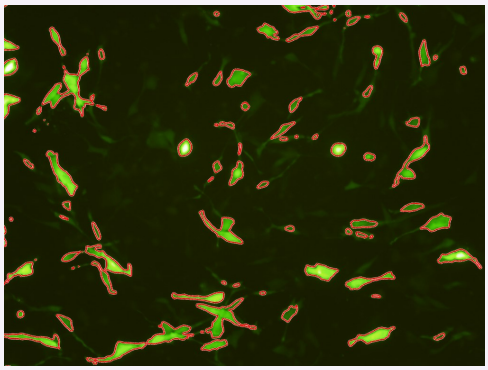


After Quantization.

- **Initial segmentation:** This is a basic segmentation. If the pixel intensity exceeds a certain minimum threshold, it is considered information (cell). Otherwise, it is considered the background. This allows obtaining the first number of cells, reducing the processing time of the final segmentation, where the final number of cells detected will be adjusted.



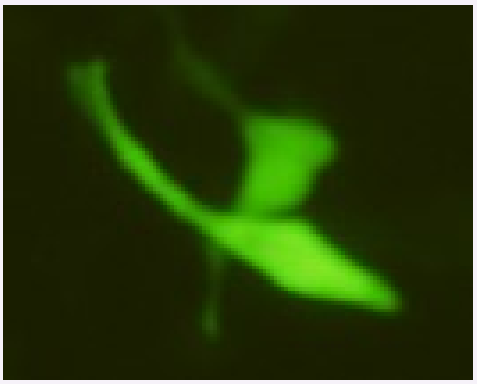
Before initial segmentation.



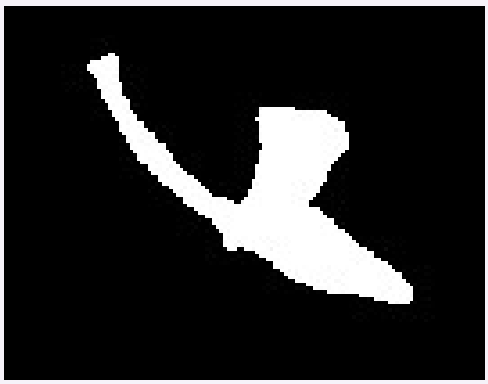
After initial segmentation.

CMSM Process (cont.)

- **Get the mask:** For each of the cells, you get your mask. From it, important parameters are obtained such as the area, the orientation, the eccentricity, the position,...

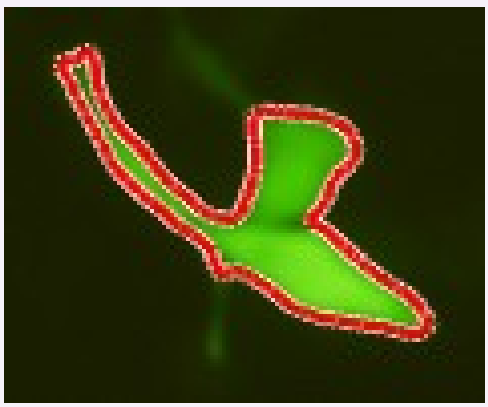


Cell.

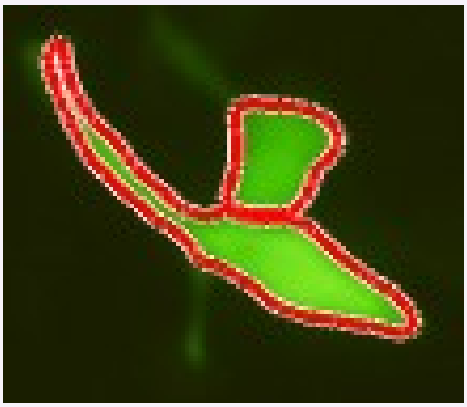


Mask.

- **Get the mask properties:** This stage is necessary to apply the CMSM. If there are several aggregate cells, that were detected as a single one in the initial segmentation, are detected and the morphology of each one is corrected to maximize the area.
- **Final segmentation (CMSM):** For each of the cells, the properties of the mask are obtained. From the quantization levels, aggregate cells are detected. To discard detected elements from the same cell, for example, the kernel and axon have been detected. It compares the orientation of each one of them. If the difference in orientations of the two objects is less than 30%, both detected objects are considered to be part of the same cell. Otherwise they are detected as distinct cells and the initial mask is divided into two, maximizing the area.



Before CMSM.

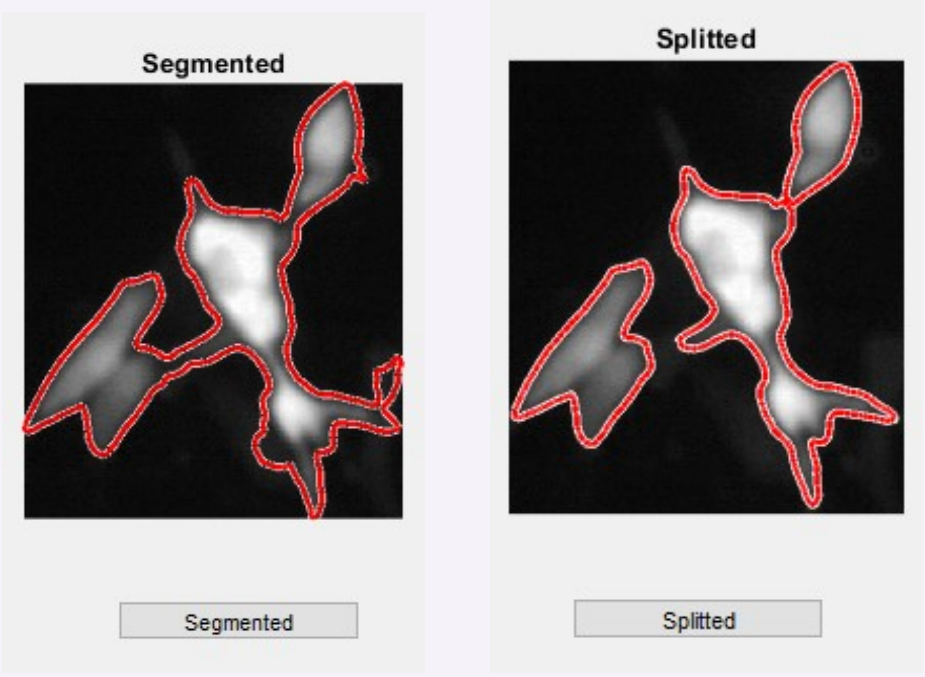


After CMSM.

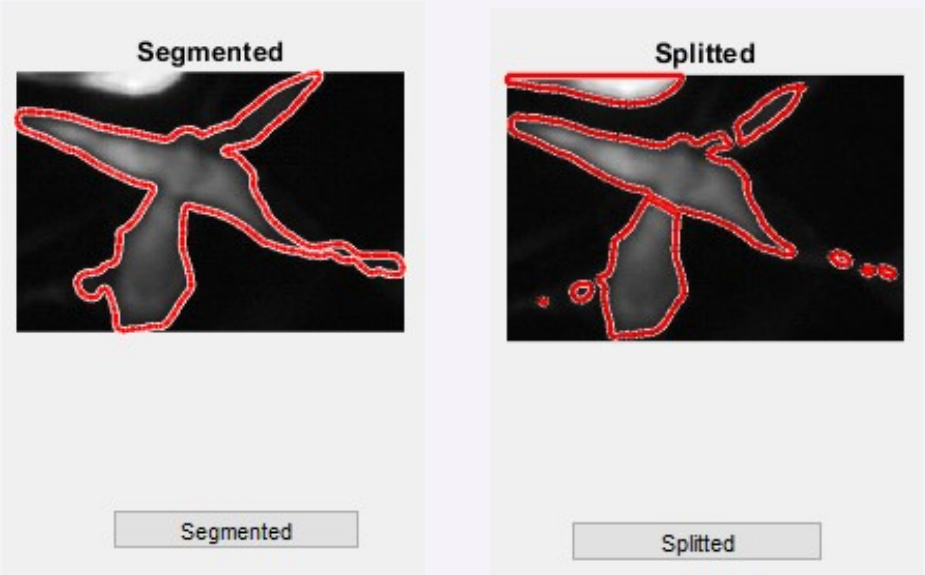
CMSM Process (cont.)

- **User Decision:** This means that the user is presented with two different segmentation results. On the one hand, the result of the initial segmentation is shown and on the other, the result of the CMSM segmentation. It is the user who decides which is best suited to the morphology of the cell.

  Note: User intervention is required only for those cells whose initial segmentation is different from that obtained by CMSM.



Initial segmentation. CMSM.



Initial segmentation. CMSM.

Figure 2.7. *Process followed by CMSM.*

3 SOFTWARE APPLICATION

The purpose of implementing software is to serve as a user interface between the researcher who uses it and the code implemented in MATLAB for the treatment of the image (see figure 1.4b). The use of this interface also allows simplifying the parameters of the segmentation.

The process of developing begins by recognizing the needs of the problem, what are the configuration parameters, and what are the functions of each one of the modules that we want to implement.

3.1 Input parameters

We want the **minimum threshold** of intensity to be the only parameter configurable by the investigator. This parameter is used to discern what is information (cells) and what is not (background). See Figure 2.6.

3.2 Requirements

As software requirements, it is necessary to perform a **segmentation** appropriate to the type of image to be treated. Besides, from the MSLab has been asked to add a module that offers the possibility of **splitting a segmented element as a single cell in two**, and another for its opposite case (**converting two segmentations into a single cell**). The last module is reserved for **data exportation** in a file after the correct segmentation has been done.

With these requirements, most of the limitations described in *1.3 Context and Problems* are solved. See Figure 3.1.

3.3 Improvements

In addition to the previous modules, the author of this work has decided to implement a series of modules that endow this tool with more versatility. For example, it has been decided to introduce a module to improve the contrast of the images.

MSLab - Problems and Limitations

- ~~The process requires knowledge on image processing.~~
- ~~The information extracted from images are restricted to intensity (and size in a few cases).~~
- ~~To get the results is necessary the intervention of the researcher.~~
- ~~The process is manual. It introduces human errors.~~
- ~~It gets long time if the number of cells are high.~~
- Sometimes, the contrast of images are very poor.
- ~~The use of MATLAB is restricted to some countries and the license is really expensive.~~
- ~~Once the information is gathered, the file format with the information depends on the researcher.~~
- The information has to be gathered image by image.

Figure 3.1. *Problems and limitations of Figure 1.3 solved by the software application.*

It has been observed on certain occasions that the reaction of the cells with a specific contrast agent is weak, which translates into a relatively low intensity, precluding an accurate segmentation.

It has also been decided to introduce modules to isolate each one of the channels of a laboratory image. On certain occasions, several contrast agents are mixed with observing certain events. If these images are treated without isolating the channels there is an interference of intensity between processes of different nature, for that reason, it has been thought advisable to add them.

Finally, it has been decided to implement a module for the representation of the data obtained when applying CMSM. This module offers graphics in a fast way that give an idea of the results obtained and indicate to the investigator if the segmentation has been done correctly.

3.4 Programming language and IDE

Since the MSLab PCs all have a version of *Windows* as operating system, and it is anticipated that futures also work in that environment, has chosen a **C#** programming language developed by the same company, so we ensure a perfect integration With the operating system of the computers of the MSLab and to obtain improvements.

To work with C#, an IDE has been chosen (**Microsoft Visual Studio**) that offers a very comprehensive tool for developing the software, debugging errors, and making publications.

Input Parameters

- Minimum intensity threshold.

Figure 3.2. *Input parameters of the software application.*

Requirements

- Suitable segmentation.
- Splitting cells.
- Merging cells.
- Exporting data.

Figure 3.3. *Requirements of the software application.*

Improvements

- Contrast enhancement.
- Gray scale filter.
- Red Channel filter.
- Green Channel filter.
- Blue Channel filter.

Figure 3.4. *Improvements of the software application.*

For each one of the modules, we will implement a different form (panel). They can be accessed from the main form of the program.

3.4.1 Main Panel

From the main panel (see Figure 3.5) we can access all the modules that will deal with the images. Some of them open a new panel. Others simply apply the filters to the image See Figure 3.7.

Adding new project

To add images to the project, press the *"Ctrl" + "N"* keys. A new window opens. There, we can add images from different locations and change the default project name and its location. See Figure 3.6.

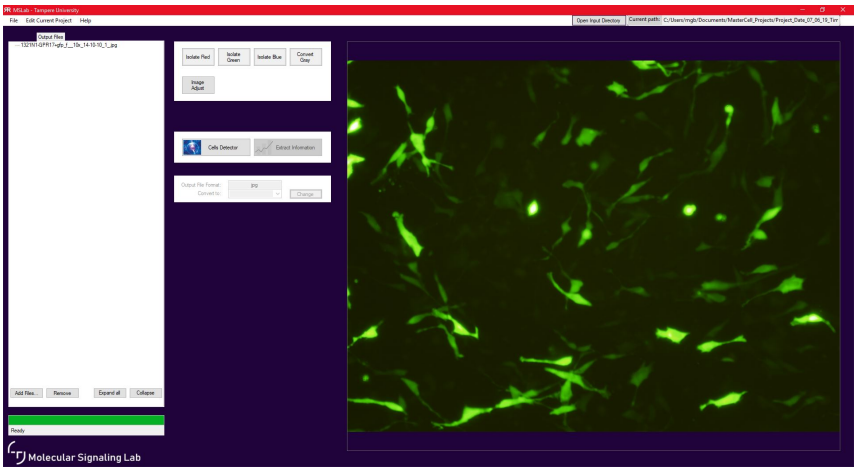


Figure 3.5. *Main Panel of MSLab software application.*

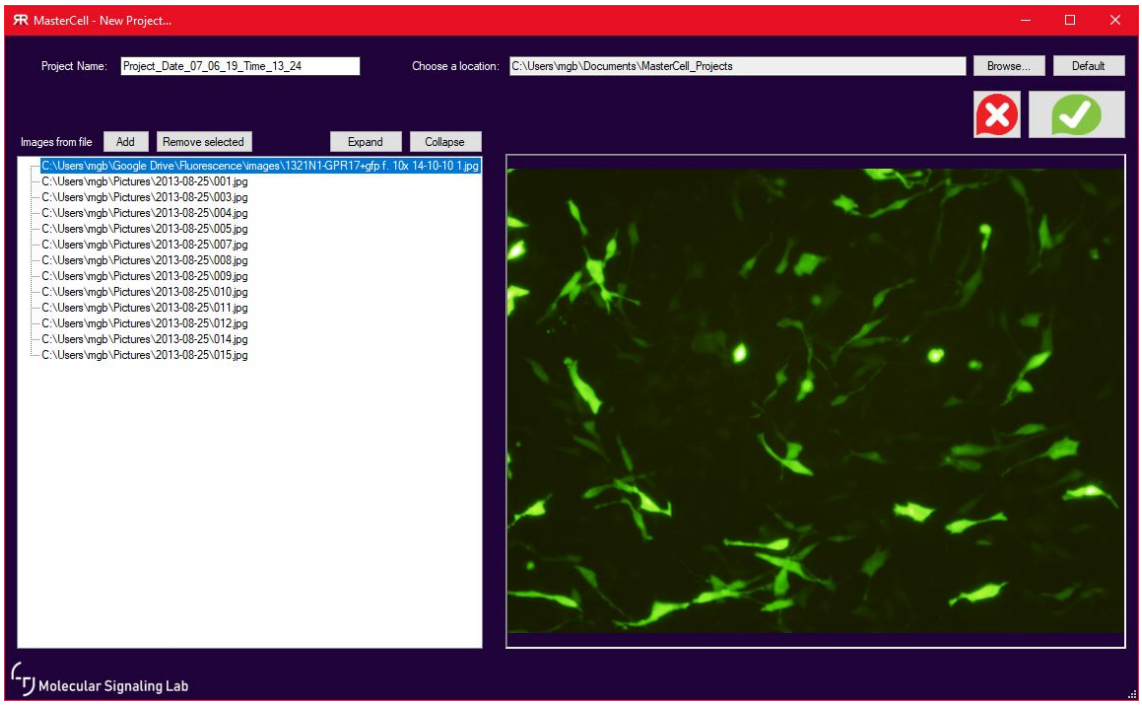


Figure 3.6. *New Project Panel.*

Channel Isolation Modules, Gray Scale module and Image Format Converter

These modules apply to the image directly without having to configure any input parameters. **Image Format Converter** is useful to store heavy images in other image format without losing intensity information.

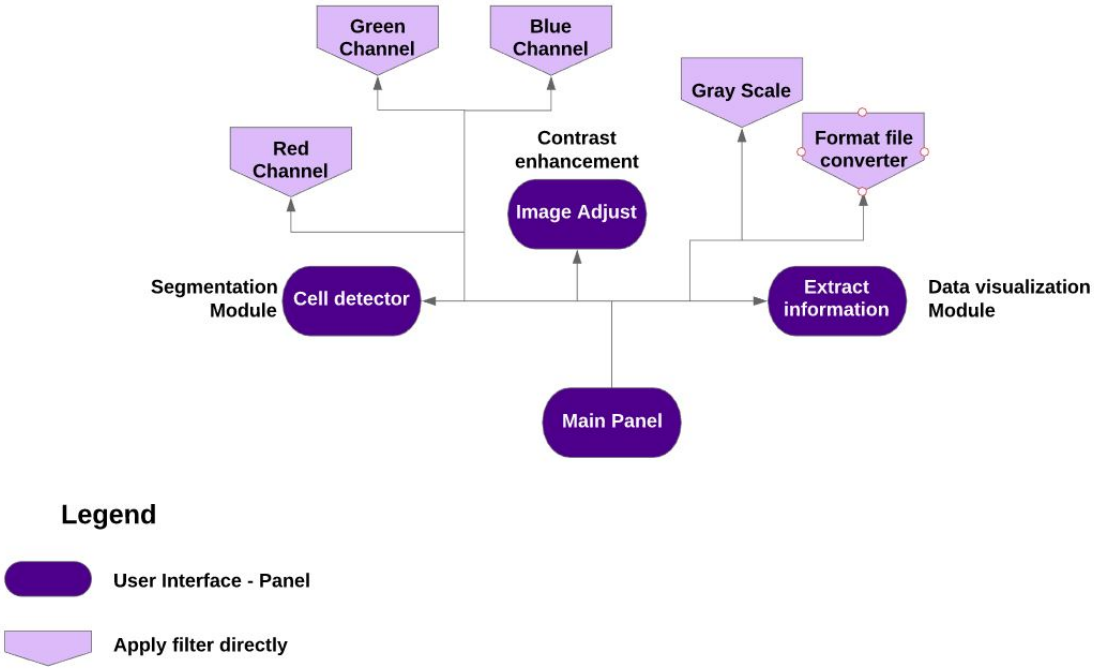


Figure 3.7. *Diagram of MSLab software application.*

Improvements

- Added functionality that allows you to work with multiple images in the same project.

Figure 3.8. *Improvements in Main Panel.*

3.4.2 Cell Detector Panel

This module is where segmentation is performed (see Figure 3.9). To do this, set the **minimum threshold** value that determines that it is considered as background and what is considered as information (cells).

Note: The optimal value of this parameter depends heavily on the quality and intensity of the image so that it can vary considerably from one image to another within the same project.

After setting the suitable value and pressing *"Detect"* button, the CMSM is applied over the image previously selected. A list of recognized elements (cells) appears and some of its characteristics such as ID or the area. From the list, we can select or deselect depending on whether we want specific elements to appear in the final image. This is necessary for example when removing intensity artefacts (very high-intensity points that are generated by the light reflection on the camera lens) whose size is the same order of magnitude of the cells. If we want to filter by the size of the detected particle, we can use the filters. To do this, activate the *"Size Filter"* button. See Figure 3.10.

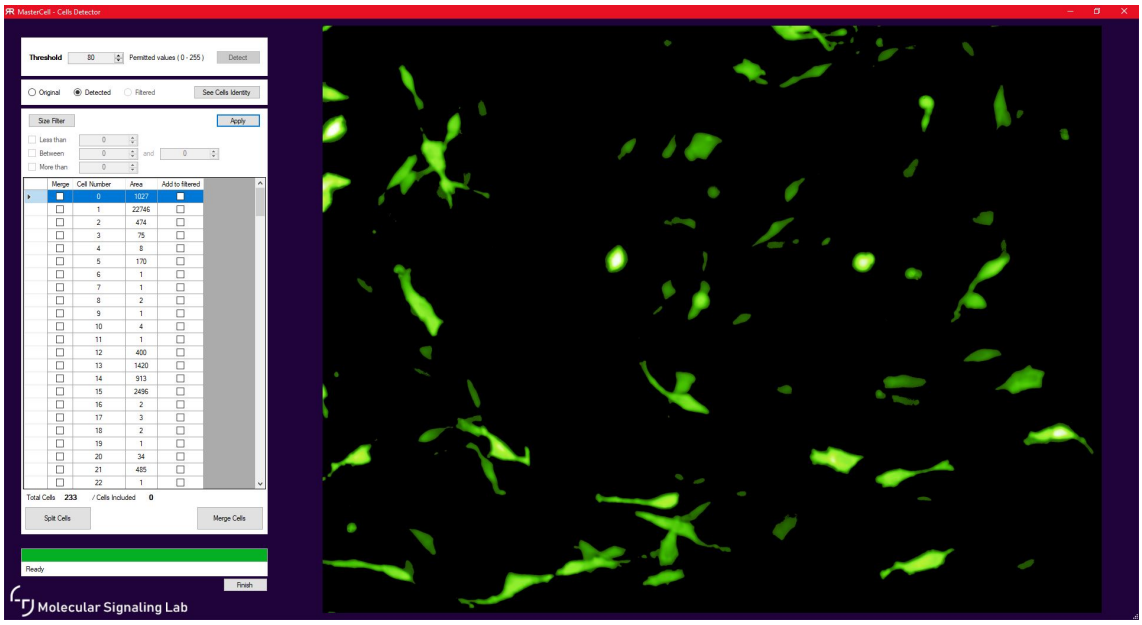
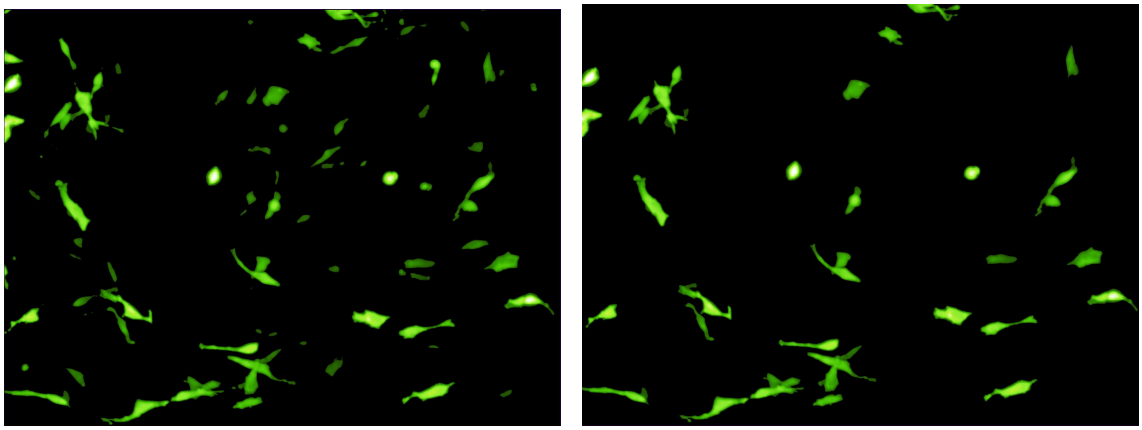


Figure 3.9. *Cell detector Panel (segmentation module) of MSLab software application.*



(a) *CMSM segmented image.* ***(b)*** *Filtered cells.*

Figure 3.10. *Filtering result of the segmentation in Cell detector Panel.*

Split Cells Panel

A new panel opens from the *"Split"* button. At the top are the coordinates of the points that will establish the separation line between the two cells that have been segmented as a single. See Figure 3.11.

After the partition is done, the CMSM is re-run on the new updated image, since the number of cells in the image has increased. See resutls in Figure 3.12

Merge Cells Panel

To join two cells, we must identify his ID. To do this, press the button *"See Cell Identity"*. The image that appears corresponds to the IDs located in the position of each cell. Once

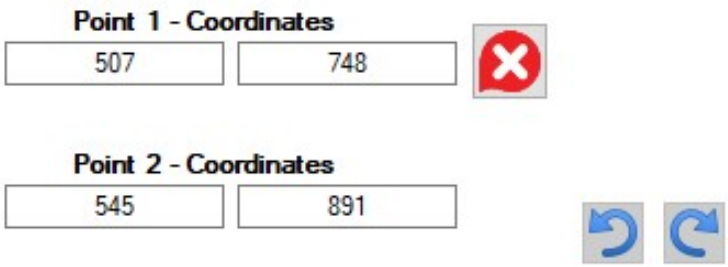
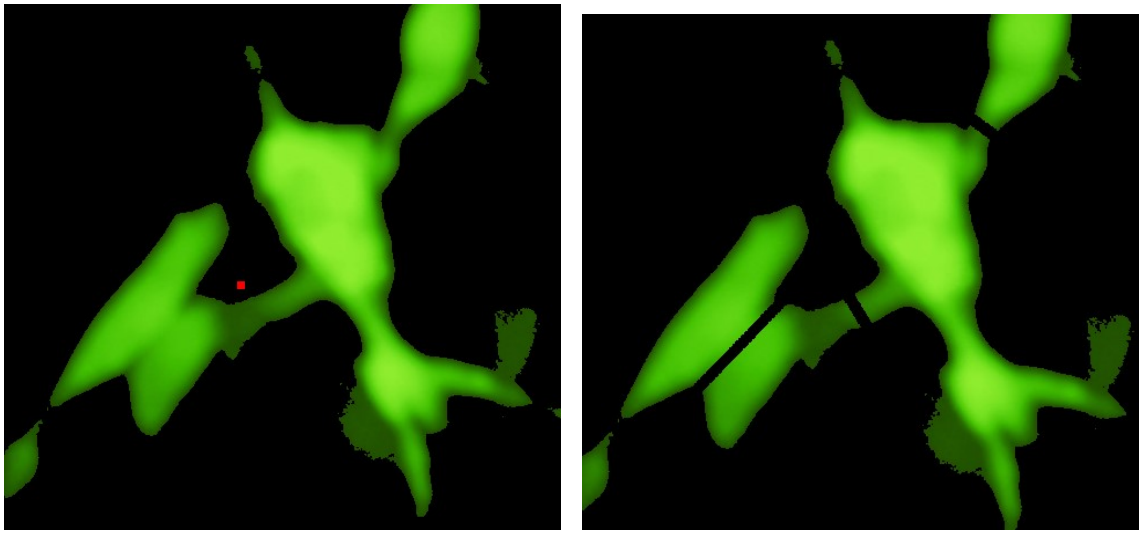


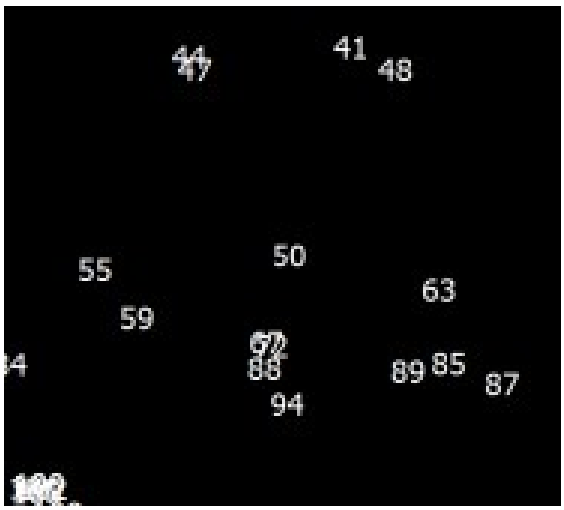
Figure 3.11. *Separation line coordinates of Split Panel.*



(a) *Before cell splitting.* ***(b)*** *After cell splitting.*

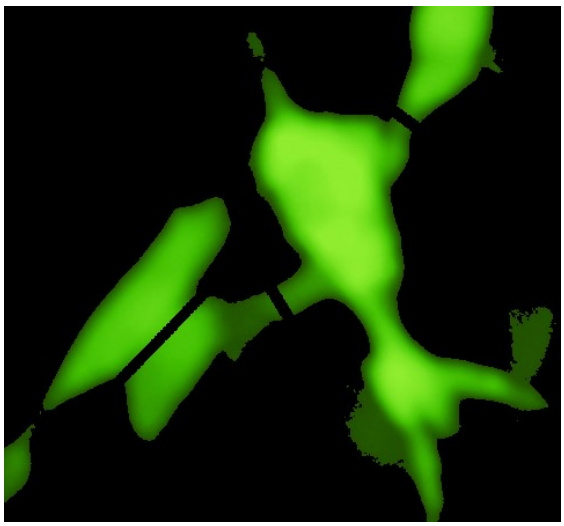
Figure 3.12. *Filtering result of the segmentation in Cell detector Panel.*

the ID is identified, they are selected in the list under *"Merge"* and after that the *"Merge cells"* button must be pressed. Again, the CMSM is re-run and the information is updated. See Figure 3.13.

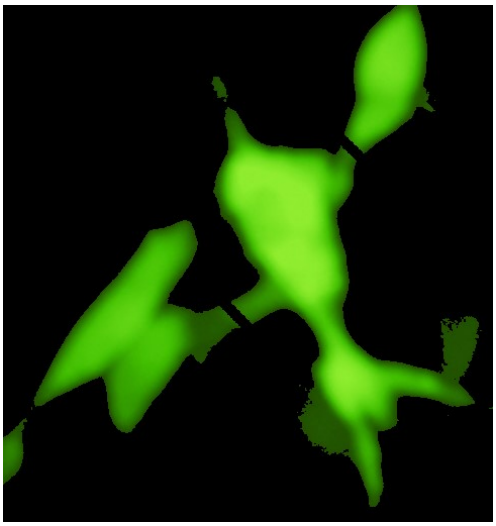


Merge	Cell Number	Area	Add to filtered
☐	54	4661	☐
☑	55	10466	☑
☐	56	3968	☐
☐	57	1	☐
☐	58	1	☐
☑	59	6912	☐

(a) *ID cells.* ***(b)*** *Merging list.*



(c) *Before cell merging.*



(d) *After cell merging.*

Figure 3.13. *Merging process in Cell detector Panel.*

3.4.3 Extract Information Panel

This is where the information obtained from the images that have been applied CMSM (left panel) is represented. When the *"Start"* button is pressed, the information appears in three graphs (see Figure 3.14):

Figure 3.14. *Extract Information Panel.*

- **Intensity Information (per cell):** The X-axis represents each of the supported intensity values. Since we work with 256 levels of intensity, the graduation of the axis goes from 0 to 255. On the Y-axis represents the number of pixels in the cell that has each intensity value. The red vertical cursor represents the average intensity value of the pixels in the cell.

  Note 1: The intensity values will always be higher than the value of the mini-

mum intensity threshold set for segmentation.

Note 2: The intensity values for each cell in each image can be known through the drop-downs.

- **Intensity Mean (per frame):** The X-axis represents the value of the ID of each cell. For each image, the maximum value of the axis will vary depending on the number of cells detected in each image. In the Y-axis, the values of average intensities are represented. The red horizontal cursor represents the average intensity of all the cells in the image.

- **Probability Distribution:** The X-axis represents the IDs of each of the cells in the image. On the Y-axis, the probability density function of the average values of all the cells in the image.

Excel File Exportation

Pressing the "Export to EXCEL" button generates a file with all the information represented in the graphs. The exportation of the data allows to share the results obtained in another PCs.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Project Nam	Project_Date_07_06_19_T			Date:	6/7/2019										
2	Project Path:	C:\Users\mgb\Document			Hour:	7:09 PM										
3																
4																
5	Probability Info															
6	Cells total in	233														
7	Cells in this i	233														
8																
9																
10	Image Info					Cells Info										
11	Image id:	0				Cell id:		0	1	2	3	4	5	6	7	8
12	Image name	1321N1-GPR17+gfp_f__10x_14-10-10_1_.jpg				Pixels:		47	206	21	18	0	36	0	0	0
13	Cells in this i	233				Intensity Mean:		167	207	107	83	0	82	0	0	0
14	MCIM (Mean	91														
15																
16	MCIM Values					Intensity Values										
17	0	0.5193133						104	80	86	81		81			
18	1	0						111	83	92	83		81			
19	2	0						119	86	97	83		80			
20	3	0						126	90	103	84		81			
21	4	0						134	94	106	84		81			
22	5	0						141	98	111	85		82			
23	6	0						150	103	114	85		82			
24	7	0						157	109	119	86		82			
25	8	0						167	114	120	86		84			
26	9	0						175	120	122	85		84			
27	10	0						184	126	122	86		84			
28	11	0						190	132	120	86		84			
29	12	0						196	139	120	86		84			
30	13	0						198	143	119	84		83			
31	14	0						201	150	115	82		84			
32	15	0						203	155	111	82		84			
33	16	0						205	160	107	81		84			
34	17	0						208	166	103	80		84			
35	18	0						208	170	99			83			
36	19	0						208	175	91			83			
37	20	0						211	180	84			83			
38	21	0						212	184				83			
39	22	0						211	187				82			

Figure 3.15. *Exportation file.*

3.4.4 Image Adjust Panel

This module allows to adjust, improve or reduce the contrast of the image. It is advantageous to work with low-intensity images or images whose intensity is quite high.

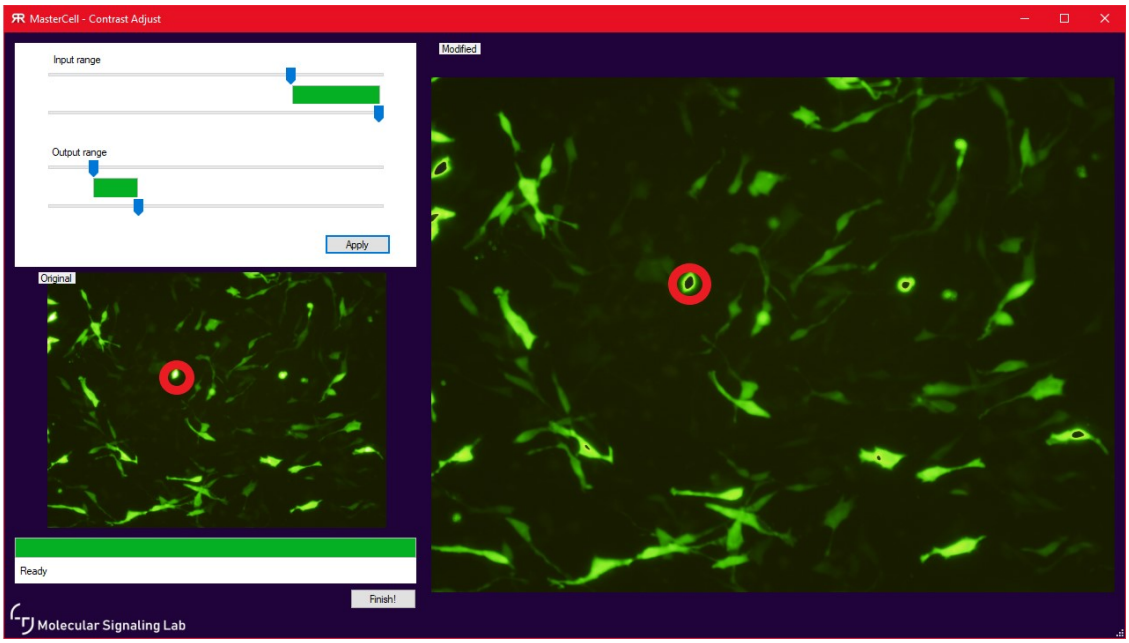


Figure 3.16. *Image Adjust Panel.*

3.4.5 Loading Panel, Name and Logo

The name of **"MSLab"** has been chosen for the software and the application logo also has been designed. See Figure 3.17.

Figure 3.17. *Loading Panel and Logo of MSLab software application.*

4 PROJECT TIMELINE

This thesis begins with the approval of the University during October 2018. Previously, the work that will be developed during the whole project is organized with the supervisor of the project, the professor Meenakshisundaram Kandhavelu.

- **Thesis start:** In this stage, we look for related information, we establish the characteristics of our project and what tools are going to be used to implement them. Information is also sought from other projects that have been implemented before, and field work is done with researchers to identify their needs and to choose how to supply them.

- **MATLAB Code stage:** Here, we implement the method using MATLAB. Only inputs from image and outputs of the segmentation are considered.

- **C# Code stage:** We start to develop the desktop application. The data visualization module is entirely implemented here.

- **Integration stage:** The segmentation code is encapsulated into the software application. Some errors are debugged and fixed. Here, the student considered to implement the enhanced contrast module.

- **Revision stage:** The final version of the software is shown to the researchers, and with the thesis supervisor, we add some final details to enhance the app.

- **Today:** The final thesis is submitted.

Figure 4.1. *Timeline of the project. From the initial steps (gathering information) to submission time (after last revision).*

5 CONCLUSIONS

To check the scope of our method, we will compare the data obtained with the CMSM with those obtained from images treated in CellProfiler.

We have done a study where both software are compared with manual counting. The **number of cells detected** for a group of 15 different images are shown in Table 5.1.

5.1 Number of cells detected

Fifteen different images have been chosen for this study. From each of them, a manual count has been performed to obtain the actual number of cells. From there, each image has been treated in each of the software of the study. Segmentation has been performed, and after detecting several cells, it has been counted for each case. The data collected are presented in the following table:

Image Number	Software	Total cells	Cells detected	Percentage (%)
0	CMSM	96	77	80
	CellProfiler		49	51
1	CMSM	105	100	95
	CellProfiler		71	67
2	CMSM	84	73	87
	CellProfiler		44	52
3	CMSM	76	70	92
	CellProfiler		43	56
4	CMSM	49	45	91
	CellProfiler		31	64
5	CMSM	65	54	83
	CellProfiler		38	59
6	CMSM	69	55	79
	CellProfiler		37	54
7	CMSM	51	35	69
	CellProfiler		23	46

Image Number	Software	Total cells	Cells detected	Percentage (%)
8	CMSM	48	40	84
	CellProfiler		26	54
9	CMSM	37	36	96
	CellProfiler		23	63
10	CMSM	26	21	81
	CellProfiler		15	58
11	CMSM	67	53	79
	CellProfiler		38	57
12	CMSM	87	73	84
	CellProfiler		53	61
13	CMSM	63	59	93
	CellProfiler		40	64
14	CMSM	39	33	83
	CellProfiler		23	58

Table 5.1. *Number of cells detected for CMSM and CellProfiler.*

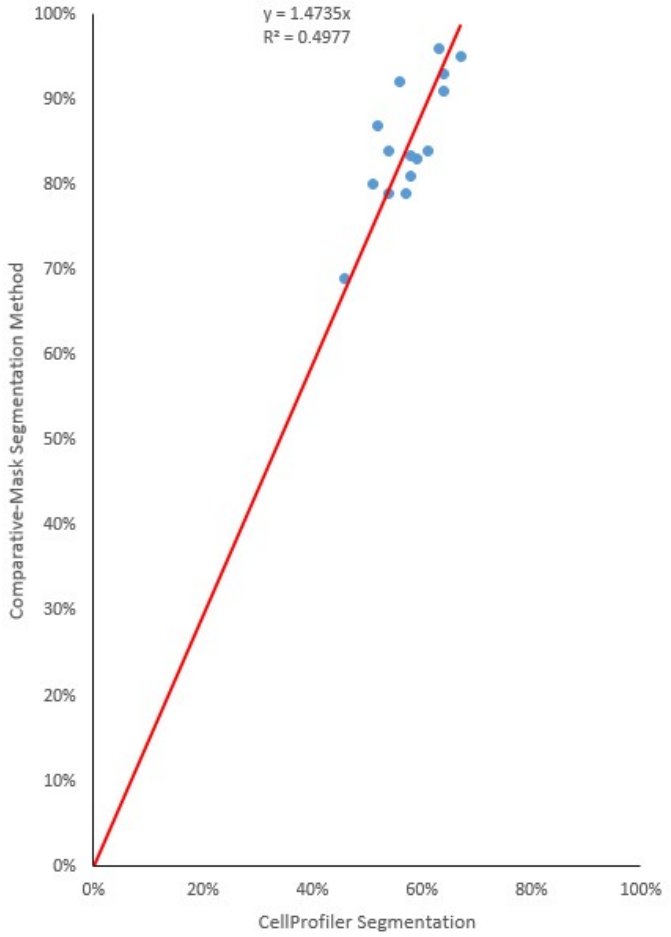


Figure 5.1. *Scatter plot of cell number for CMSM and CellProfiler.*

Conclusions

- The results show that the CMSM performs a better segmentation, since the success rate is greater than in the case of the CellProfiler.
- Sometimes using general-purpose tools may be a good idea if we do not have the tools we need more specific or enough time to implement them. However, when we can have specific software for the task done, the results improve.

Figure 5.2. *Conclusions of the comparison.*

5.2 Project Improvements

- A significant improvement could be to extend segmentation to another type of cell, not just neuronal. This would improve the versatility of the application and would raise positions when it was used, not only by the MSLab, perhaps it could be considered to distribute it to more departments of other universities.

- Nowadays, machine learning is being used in many areas of image processing since it considerably improves the results obtained. As an improvement, the possibility of including it is considered.

- Currently, the software architecture is 64 bits. This restriction is imposed by MATLAB because this software has this same architecture. Extending the execution to 32-bit machines can be a significant improvement point if one takes into account that many of the lab computers use an already old 32-bit architecture.

REFERENCES

[1] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland and D. M. Sabatini. "CellProfiler: image analysis software for identifying and quantifying cell phenotypes". *Genome BIology* 7.10 (2006).

[2] S. Chowdhury, M. Kandhavelu, O. Yli-Harja and A. S. Ribeiro. "Cell segmentation by multi-resolution analysis and maximum likelihood estimation (MAMLE)". *BMC Bioinformatics* 14.10 (2013), 2–4.

[3] E. Hodneland, T. Kögel, D. M. Frei, H.-H. Gerdes and A. Lundervold. "CellSegm - a MATLAB toolbox for high-throughput 3D cell segmentation". *Source Code for Biology and Medicine* 8.16 (2013).

A MATLAB CODES

A.1 Comparative-Mask Segmentation Method

```
clc;
clear variables;
close all;

% INPUT Level of quantization image
Q = 20;
threshold_min = 3;
%i = 17;

% Load image
[imgC, height, width] = readImage('images/1321N1-GPR17+gfp f. 10x 14-10-10 1.jpg');
% A conversion to Gray Scale is needed
imgG = fromRGB2Gray(imgC);
imgG = wnrFilter(imgG);
imgG = openImage(imgG,'disk',100);
imgG_quantized = quantizeImage(imgG,Q);
imgG_mask = imgG_quantized > threshold_min;
figure, imshow(imgC,[]), hold on, visboundaries(imgG_mask,'Color','r'), hold off, title('Segmented')
[cc,info] = objectsProperties(imgG_mask);
Object_Box = getImageObjects(cc,info,height,width);
modifiedFromOriginal = [];
numberCellsBefore = cc.NumObjects;
index = 1;

for i = 1 : cc.NumObjects
    initialMask = getObjectMask(imgG_mask,info,i);
    [ccInitialMask,infoInitialMask] = objectsProperties(initialMask);
    numElemBefore = length([infoInitialMask(:)]);

    imgAux = getImage(imgG_quantized,Object_Box,i);
    finalMask = segmentDifferentObjects2(imgAux);
    [ccFinalMask,infoFinalMask] = objectsProperties(finalMask);
    numElemAfter = length([infoFinalMask(:)]);

    if numElemAfter > numElemBefore
        objectFromOriginal = getImage(imgC,Object_Box,i);
        objectFromSegmentation1 = getImage(initialMask,Object_Box,i);
        objectFromSegmentation2 = finalMask;
        result = showBothResults(objectFromOriginal,objectFromSegmentation1,objectFromSegmentation2,index);
        if result == 2
            modifiedFromOriginal = [modifiedFromOriginal i];
            imgG_updatedMask = updateMask(imgG_mask,i,info,finalMask);
            imgG_mask = imgG_updatedMask;
        end
        index = index + 1;
    end
end
[ccFinalMask,infoFinalMask] = objectsProperties(imgG_updatedMask);
numberCellsAfter = ccFinalMask.NumObjects;
```