FEDERAL UNIVERSITY OF SANTA CATARINA
SCHOOL OF TECHNOLOGY
DEPARTMENT OF AUTOMATION AND SYSTEMS
UNDERGRADUATE PROGRAM IN CONTROL AND AUTOMATION ENGINEERING

Rafael dos Santos Peixoto

Enhancing the resolution of spatial transcriptomics through morphology analysis

Florianópolis
2022

Rafael dos Santos Peixoto

Enhancing the resolution of spatial transcriptomics through morphology analysis

Final report of the discipline DAS5511 (Final Work Project) as Final Paper do Undergraduate Program in Control and Automation Engineering da Federal University of Santa Catarina em Florianópolis.
Supervisor:: Prof. Edroaldo Lummertz da Rocha, Dr.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Peixoto, Rafael
Enhancing the resolution of spatial transcriptomics through morphology analysis / Rafael Peixoto ; orientador, Edroaldo Lummertz da Rocha, 2022.
77 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2022.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Spatially resolved transcriptomics. 3. 10X Visium. 4. Nuclei segmentation. I. Lummertz da Rocha, Edroaldo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Rafael dos Santos Peixoto

Enhancing the resolution of spatial transcriptomics through morphology analysis

This monography was judged in the context of the discipline monography (Final Work Project) and **APPROVED** in its final form by the Undergraduate Program in Control and Automation Engineering

Florianópolis, March 29, 2022.

Prof. Hector Bessa Silveira, Dr.
Program Coordinator

Examination Board:

Prof. Edroaldo Lummertz da Rocha, Dr.
Advisor
UFSC/CCB/MIP

Prof. Edroaldo Lummertz da Rocha, Dr.
Supervisor
UFSC/CCB/MIP

Luis Fernando Nazari, Dr.
Evaluator
IFC - Rio do Sul

Prof. Eduardo Camponogara, Dr.
President of the Examining Commission
UFSC/CTC/DAS

I thank God and Our Lady of Aparecida for giving me the opportunity and conditions to develop this project, and I dedicate it to my dear parents, my sister, and friends.

ACKNOWLEDGEMENTS

I thank professor Edroaldo who introduced me to bioinformatics. His efforts and patience made me develop both professionally and personally. During my time working with him, I changed from someone ignorant in biology to a student eager to continue researching in the area.

I also thank professor Ruben, who accepted the challenge of mentoring a student from another country. His assistance guided me in the field of spatial omics and gave me a broader perspective about the topic, which will be essential for succeeding in graduate school.

“For comprehensive molecular understanding of tissues, spatial transcriptomics methods aim to characterize gene expression profiles while retaining information of the spatial tissue context (BURGESS, 2019).”

ABSTRACT

This work intended to enhance the resolution of Spatial Transcriptomics datasets by assigning the cell type to its location in the tissue. To perform this task, we focused on segmenting the cells, extracting their morphological features, and creating a classifier from them. The segmentation used the Mesmer algorithm as its core function and performed well in both 4',6-diamidino-2-phenylindole (DAPI) and haematoxylin and eosin (HE) images. The morphological information was extracted using the Fiji software, but due to its characteristics, it could not be integrated into our R package, hindering the development of the classification task. The quality of the obtained features from the cell masks and the lack of other packages to perform the task showed the opportunity to develop new projects in the area. The quality of the segmentation demonstrates the robustness of the method and indicates that it could also be applied to other projects. Therefore, it was incorporated into the Giotto package and made open source. Additionally, the results presented here give clear guidelines to continue it in the future.

Keywords: Spatially resolved transcriptomics. 10X Visium. Nuclei segmentation.

RESUMO

Este trabalho teve como objetivo melhorar a resolução de dados de Spatial Transcriptomics, atribuindo o tipo de célula à sua localização no tecido biológico. Para realizar essa tarefa, focamos em segmentar as células, extrair suas características morfológicas e criar um classificador a partir delas. A segmentação usou o algoritmo Mesmer como sua função principal e teve um bom desempenho em imagens DAPI e HE. A informação morfológica foi extraída no software Fiji, mas devido às suas características, não pôde ser integrada ao nosso pacote R, dificultando o desenvolvimento da tarefa de classificação. A qualidade das características obtidas das máscaras celulares e a falta de outros pacotes para realizar a tarefa mostraram a oportunidade de desenvolver novos projetos na área. A qualidade da segmentação demonstra a robustez do método e indica que também pode ser aplicado a outros projetos. Por isso, foi incorporado ao pacote Giotto e tornado open source. Além disso, os resultados apresentados aqui fornecem diretrizes claras para a continuação da tarefa no futuro.

Palavra-chave: Transcriptoma espacialmente resolvido. 10X Visium. Segmentação de núcleos.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

DAPI	4',6-diamidino-2-phenylindole
FFPE	formalin-fixed, paraffin-embedded
FISH	Fluorescence In Situ Hybridization
HE	haematoxylin and eosin
MERFISH	multiplexed error-robust FISH
scRNA-seq	Single-cell RNA sequencing
ST	Spatial Transcriptomics

CONTENTS

1 INTRODUCTION

The transcriptome, the set of RNA molecules transcribed from the DNA sequence of our cells, provides a blueprint of all genes expressed by the cells in our body. Different cell types may express a different set of genes, which are related to cell type-specific biological functions. Therefore, the transcriptome is a molecular measure of cell identity, and understanding transcriptomic differences between cell types in health and disease is important to understand tissue physiology.

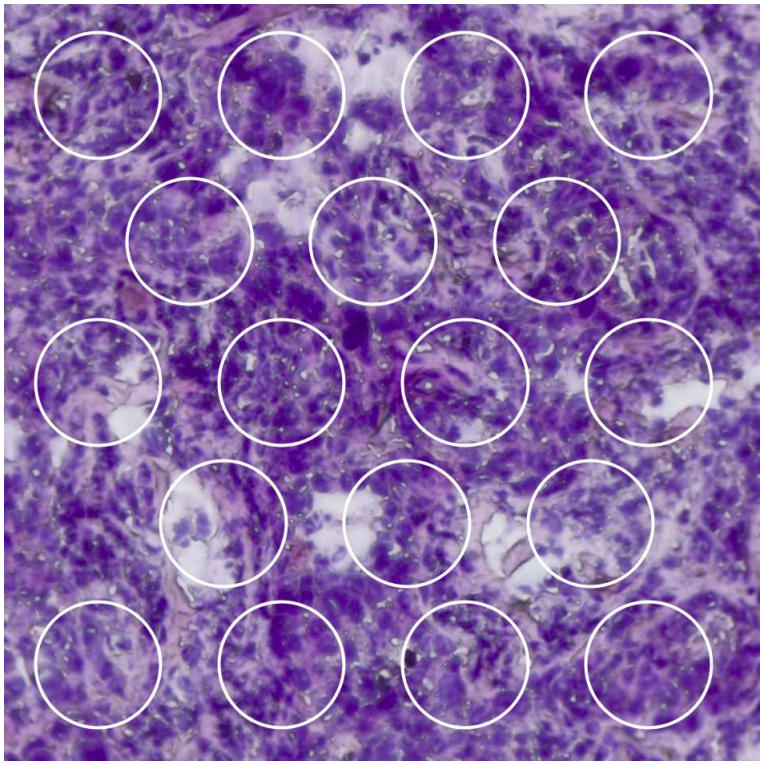
In 2013, the journal Nature Methods defined the technologies that quantify the transcriptome of individual cells - Single-cell RNA sequencing (scRNA-seq) - as the method of the year. However, due to the tissue dissociation process required to generate scRNA-seq data, the spatial localization of cells in the tissue is lost. Importantly, depending on the location of a particular cell type in the tissue, cellular behavior might be distinct. Thus, preserving the spatial information of cell types is crucial to understanding tissue architecture. Later, in 2020, the same journal stated spatially resolved transcriptomics - the ability to quantify the transcriptome while preserving the spatial location of the cells - as the method of the year. By integrating the tissue image based on histology and the spatial location of gene expression, analysis of spatial transcriptomics data is quite challenging and new algorithms are required. Interestingly, this technology provides new avenues to develop computer vision algorithms to interpret tissue images and their gene expression profiles (matrices that contain genes as rows and cells as columns), providing an unprecedented resolution to understand tissue biology.

The advance of technologies to collect transcriptomic information with spatial resolution allowed scientists to comprehend how gene expression varies spatially. These methods allowed the identification of new cell types or cell states within the tissue. Moreover, they brought new insights about the architecture of the tumor-microenvironment interface, how the distance between cells affects their communication, and also about tissue development. These techniques can be divided into two major categories depending on how they collect the data: barcode-based methods and image-based ones. They are still not perfect and need to balance resolution and throughput. The ones that capture subcellular resolution are restricted to measuring only hundreds of genes, such as the MERFISH provided by the Vizgen company, while the ones that count thousands of genes capture the information of multiple cells at the same time, such as the ST provided by the Visium 10X company.

The barcoded method used in this study is the ST one. Figure 1 presents an illustration of how the barcodes of ST are placed in the tissue. The transcriptomic information of each spot is given along with the HE image. The number of cells (represented by their nuclei in purple) in each circle can vary depending on the tissue, but in general, it ranges from 5 to 10. Therefore, instead of having a matrix of genes per cell, it provides

only a matrix of genes per spot, which limits the understanding of tissue diversity.

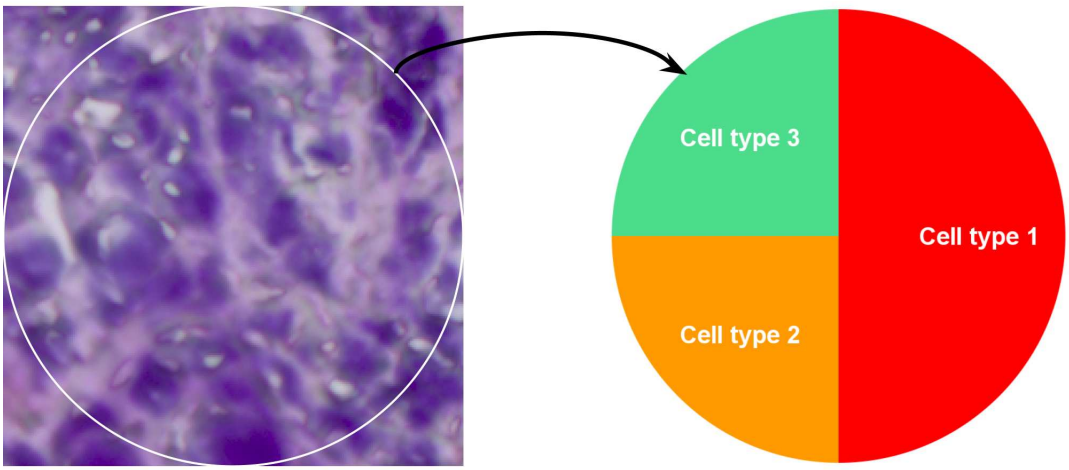
Figure 1 – Illustration of the spots (circles) in ST.



Source: Personal archive.

Today, there are tools to calculate the percentage of cell types in each ST spot (Figure 2). Yet, they do not assign the type to the cell location and only present a graphical representation of the deconvolution. A method with high throughput and resolution would allow a deeper understanding of tissue diversity, providing new insights into how cells are affected by their location.

Figure 2 – Representation of cell-type deconvolution of ST.

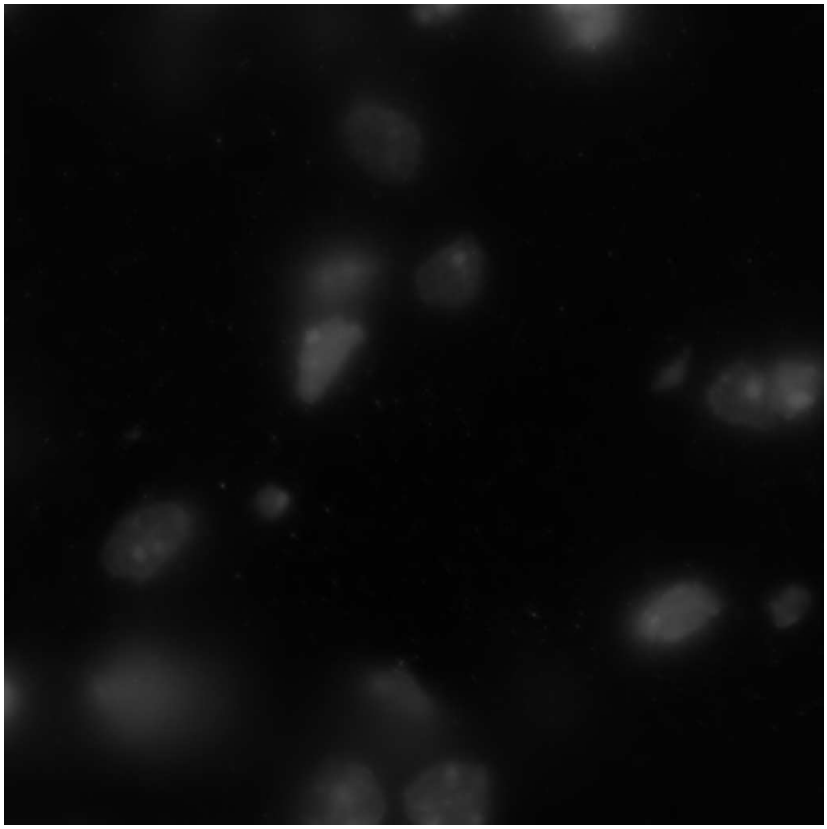


Source: Personal archive.

Therefore, the task intended in this project was to develop an algorithm that will overcome the multi-cellular resolution limitation of the ST technology by assigning cell types to the specific cell location.

As there is no ground truth for the cell types of the ST dataset, we applied this approach to another technology. The MERFISH data presents the expression and spatial information of individual cells along with a DAPI stained image of the nuclei (Figure 3). Therefore, it will be used to create a pseudo-ST dataset which will be used to validate our approach.

Figure 3 – Illustration of the transcripts in each individual cell as obtained from the MERFISH data.



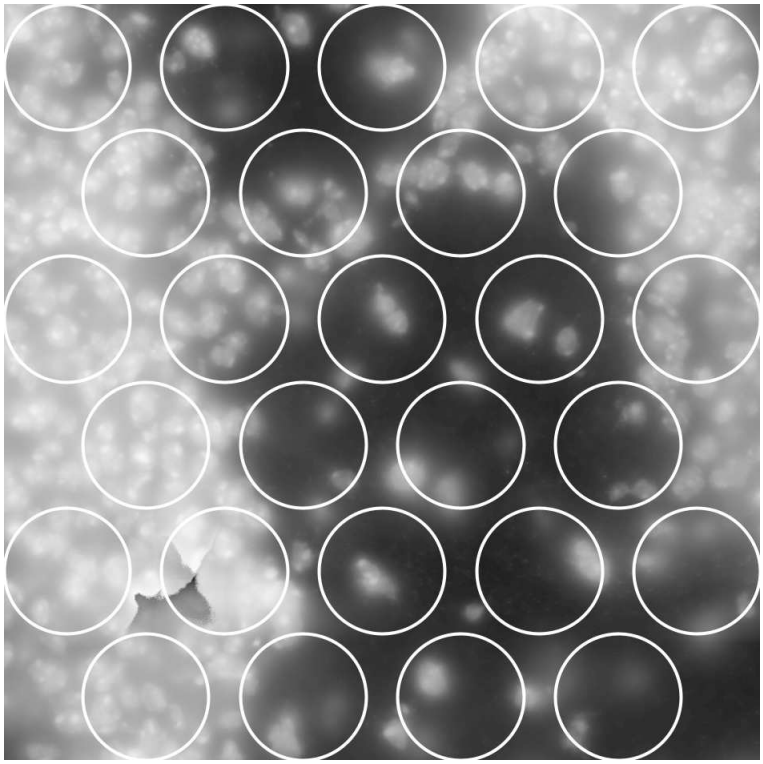
Source: Vizgen.

1.1 OBJECTIVES

The overall objective was to develop an algorithm based on computer vision and machine learning to enhance the resolution of Spatial Transcriptomics by assigning the cell type to the cell location using morphology analysis of the segmented cells in the tissue. However, as there is no labeled data for the ST dataset, we focused on testing this approach using pseudo-ST data created from the MERFISH dataset.

Therefore, the first step is to create the pseudo-ST dataset. Figure 4 illustrates this process.

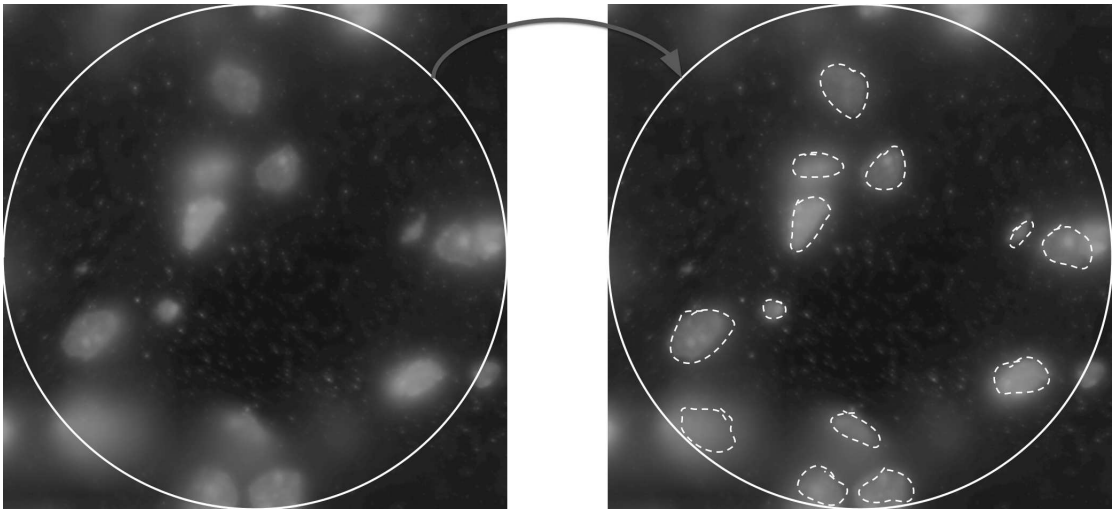
Figure 4 – Illustration of the pseudo-ST.



Source: Personal archive.

The second step is to segment cells in the associated tissue image. Figure 5 represents the ST spot before and after the cell segmentation, where the cells are the black dashed circles.

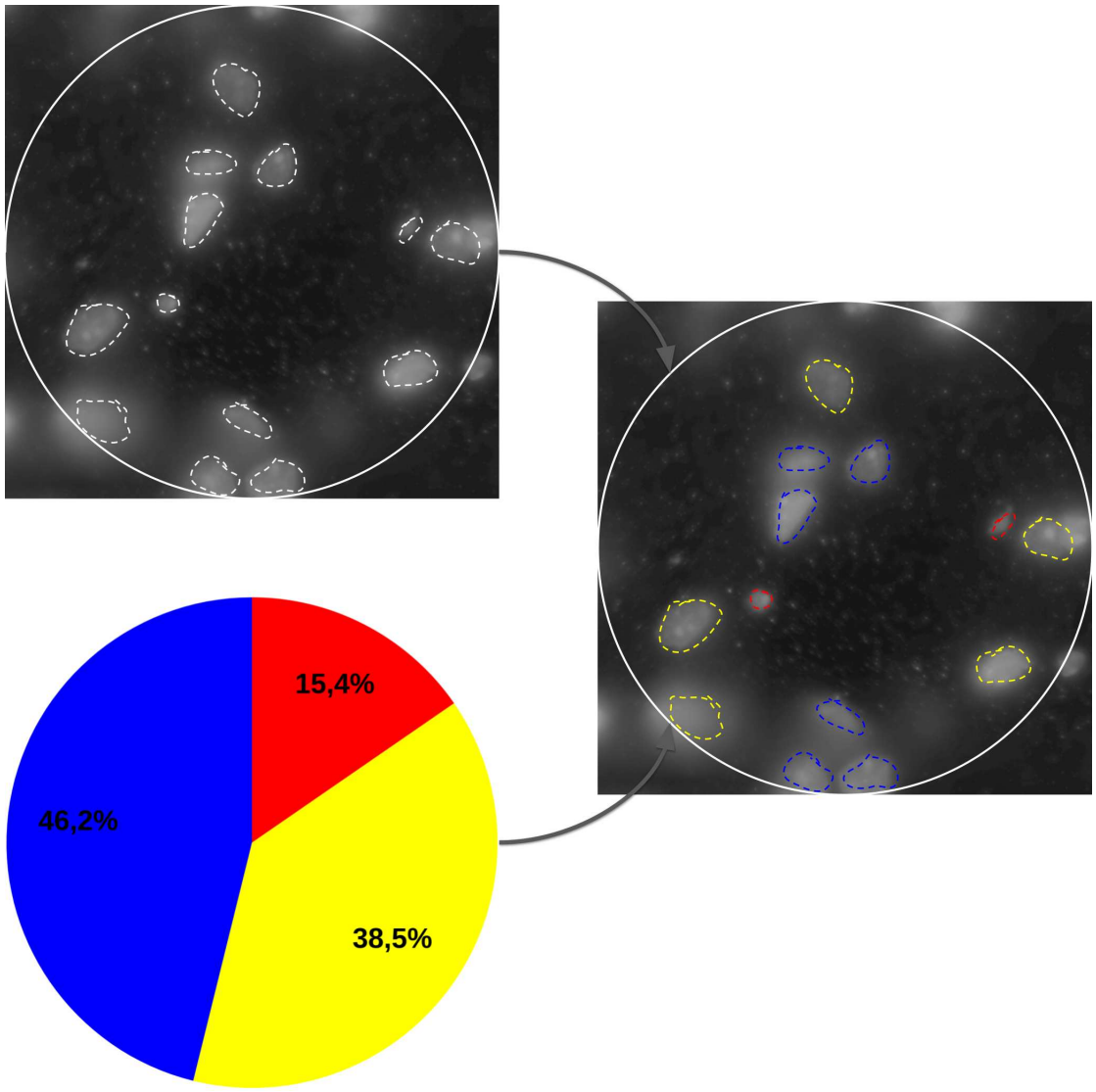
Figure 5 – Segmentation of the cells in the ST spot.



Source: Personal archive.

The third step is to assign the cell type obtained in the deconvolution to the cell location by classifying the morphology information obtained in the segmentation (Figure 6). It is worth mentioning that we chose to use morphological information because we aimed to understand which features are correlated with specific cell types. Therefore, we avoid deep learning methods due to their poor interpretability.

Figure 6 – Assignment of the cell type to its location.



Source: Personal archive.

Finally, after validating the approach, the last step is to extend it to the original ST dataset. Therefore, except for the creation of pseudo-ST, all the other steps would be repeated for the HE image and the expression matrix for the ST spots.

1.2 LITERATURE REVIEW

The methods to analyze spatial transcriptomics data are divided into five categories: spatial clustering, identification of spatially variable genes, cell-type deconvolution in spatial transcriptomics spots, enhancement of gene expression resolution, and identification of cellular interactions. The methods of spatial clustering groups similar cells or spots by their spatial and transcriptomic information. The identification of spatially variable genes aims to detect spatial patterns of gene expression. The cell-type deconvolution aims to calculate the percentage of cell types in each spot. The enhancement of gene resolution aims to reduce the size of the spots and predict the gene expression in each of them. The identification of cellular interactions intends to incorporate spatial information into the cellular communication analysis. Most of the tools available today lack to integrate image analysis with the spatial and transcriptomic information (HU et al., 2021).

This work will incorporate tools of cell-type deconvolution and image analysis to enhance the resolution of spatially resolved transcriptomics. Different from the existing methods, we will not focus on gene expression. Instead, we will aim to perform this step by working with the cell types and the segmented cells. To the best of our knowledge, there is not a work in literature that performs this task. Therefore, the accomplishment of this project will result in a novel approach to investigating cellular diversity.

1.3 DOCUMENT STRUCTURE

Chapter 2 describes the institutions in which the FWP was performed. This project was a collaboration between the Lummertz da Rocha Lab (UFSC) and the Dries Lab (Boston University). Therefore, this chapter describes both laboratories, their process, and how this work contributed to each of them. In Chapter 3, we describe the theoretical foundation that guided this thesis. We provide information on the data and also the tools used. Chapter 4 explains the general requisites, functional or not, considered in the project. It includes the major goal and the necessary steps to achieve it. Chapter 5 details the final product of the project. It explains how the methods described in chapter 3 were put together to achieve the desired result. It also explains which decisions were made to keep the development aligned with its requisites and the extent to which they were satisfied. In Chapter 6, we analyze the results, the advantages, and disadvantages of the algorithm, and its impacts on the field of spatial omics. Finally, chapter 7 concludes the work by summarizing everything that was mentioned in the previous chapters, identifying limitations in the project, and indicating future works.

2 LABORATORIES

In this chapter, section 2.1 explains the context of the collaboration between the laboratories. Then, section 2.2 describes the Lummertz da Rocha Lab, and section 2.3 the Dries Lab.

2.1 COLLABORATION BETWEEN LABORATORIES

The present work was conducted as a collaboration between the Lummertz da Rocha laboratory, part of the Department of Microbiology, Immunology, and Parasitology at the Federal University of Santa Catarina, Brasil, and the Dries Lab located in the Boston University Medical Campus and part of the Department of Hematology and Medical Oncology and Computational Biomedicine. Therefore, the project took advantage of both laboratories in combining gene expression with image analysis. This chapter describes both groups and how the project contributed to each of them.

2.2 LUMMERTZ DA ROCHA LABORATORY

By creating and implementing systems biology methodologies, the Lummertz da Rocha group hopes to better understand the cellular and molecular mechanisms underlying cell phenotypes in health and illness. This knowledge is used to create novel cell types, unravel phenotypic dysregulation in illness, and find new therapeutic possibilities.

Tissue ecosystems, stem cell engineering, systems biology, and machine learning are the primary areas of the organization. The first attempts to learn how tissue ecosystems are influenced during normal homeostasis and how they change during pathogenic processes like cancer and viral disorders. To put it another way, the goal is to analyze the cell-type makeup of tissues in different disease states and identify plausible reasons that cause the shift. The second goal is to learn how the immune system develops and use that information to create clinically relevant cell types for cell therapy. It investigates how stem cells differentiate in order to mimic this behavior in a lab setting. The third, which includes this project, is concerned with developing computer algorithms to assess biological data in order to develop data-driven hypotheses that will direct our experimental study. It tries to explain how cells communicate with one another and how their transition dynamics work.

The CellRouter algorithm was created by the team to analyze single-cell sequencing data. It's a complex single-cell analysis platform with data processing and visualization functions implemented in R. Complex single-cell trajectories are a specialty of the program. It was tested using data from single-cell RNA sequencing (ROCHA et al., 2018). The team built additional functions to analyze this data in response to the in-

troduction of new sequencing methods, such as Spatial Transcriptomics (STÅHL et al., 2016) and integrated them into the current algorithm.

Furthermore, the team just released CellComm, a platform for studying intercellular communication and how it influences cell differentiation. The aorta-gonad-mesonephros (AGM) influenced hematopoietic stem cell emergence in this study, which was also corroborated using other scRNA-seq data. This tool included the algorithms for ST analysis to incorporate the distance between cells as a factor in their communication.

The work developed previously in Rafael Peixoto's Mandatory Internship at the laboratory confirmed the importance of understanding the cell location in systems biology. Yet, it also showed how the low resolution of the ST can skew the results. Therefore, this work would contribute to the Lummertz da Rocha Laboratory by unveiling the tissue morphology and improving the previous results in cellular communication.

2.3 DRIES LABORATORY

The group focus on developing tools to take advantage of the latest advances in spatial and functional genomics, imaging, and tissue modeling. These algorithms are applied to bring new insights into cancer biology, epigenetics, and transcription. In both health and disease, the lab focus on learning more about the transcriptional principles of cellular plasticity and the sources of diversity in multicellular tissues. It aims to better understand and intervene in processes like tumor growth and treatment resistance by producing experimental data and employing computational and statistical tools. The group is particularly interested in enhancing breast cancer detection and treatment options, and it works with collaborators in the Boston Medical Center to remove racial disparities in cancer care and research.

The laboratory developed Giotto: a toolbox for integrative analysis and visualization of spatial expression data. This package is divided into two parts: analysis and visualization. The analysis module performs end-to-end analysis by utilizing a variety of algorithms for determining tissue composition, spatial expression patterns, and cellular interactions. Additionally, data from scRNA-seq can be used to analyze spatial cell-type enrichment. Users can visualize analysis outputs and imaging features interactively using the visualization module (DRIES et al., 2021). Additionally, the team is constantly updating the software to include new methods and other types of spatial omics data.

One of the algorithms incorporated in Giotto is the SpatialDWLS. This method analyses the gene expression to estimate the percentages of cell types in each spot (DONG; YUAN, 2021). The cell-type deconvolution tool allows a deeper comprehension of cell diversity, but it does not indicate the specific location of the cell types. Therefore, the current work aims to address this issue by assigning the deconvolved cell types to their specific location.

3 THEORY

In this chapter, section 3.1 describes the data used in this project. Then, section 3.2 explains the methods used in the image segmentation, section 3.3 the tools for extracting morphological information, and section 3.4 the classification.

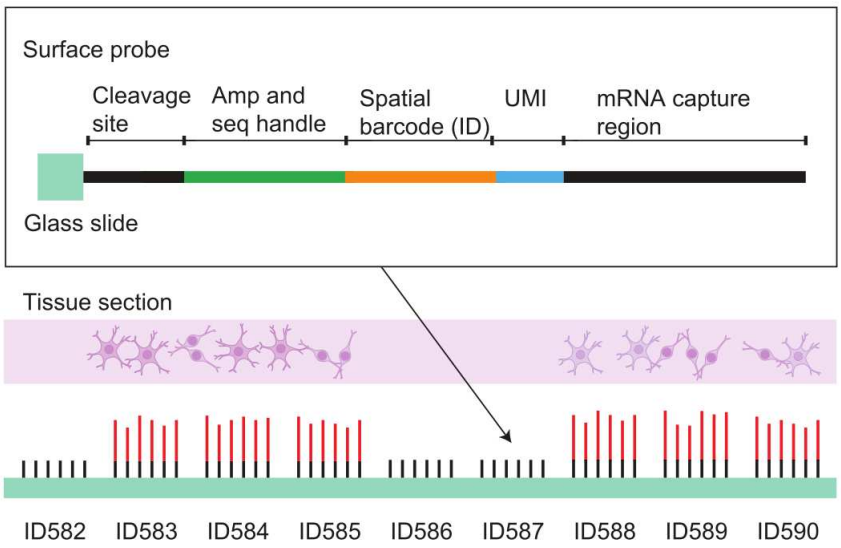
3.1 DATA

The SRT technologies are divided into image-based in situ sequencing and spatial barcoding, followed by next-generation sequencing (NGS). Subsection 3.1.1 explains the ST technology, which is based on barcoding, while subsection 3.1.2 explains the MERFISH, which is based on imaging. Finally, subsection 3.1.3 explains the pseudo-ST data.

3.1.1 Spatial Transcriptomics

In this work, we focused on the spatial barcoding approach called Spatial Transcriptomics (STÅHL et al., 2016). It works by placing identifiers at specified locations in the tissue, indexing each of them, and then using NGS to quantify the genes present in each region (Figure 7). It can be utilized on formalin-fixed, paraffin-embedded (FFPE) tissues as well as samples stained with HE.

Figure 7 – Barcoding process from ST technology.

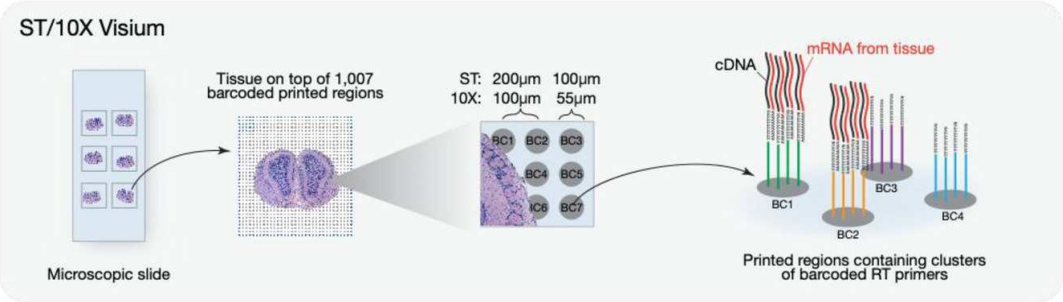


Source: Adapted from (STÅHL et al., 2016)

The ST method, first published in 2016, used NGS to quantify the transcripts. This feature allowed the reading of the complete transcriptomic information. However, due to the limitation generated in the barcoded process, the spots may contain more

than one cell, resulting in a multicellular resolution (ASP; BERGENSTRÅHLE; LUNDEBERG, 2020). The first version of the ST technology had spots of 100μm in diameter that were 200 micrometers apart from each other. The 10X Genomics corporation recently acquired the method (MARX, 2021), improved it, and created the 10X Visium. This second version made it possible to quantify gene expression within spots of 55 micrometers in diameter that are 100 micrometers apart from each other (ASP; BERGENSTRÅHLE; LUNDEBERG, 2020). Figure 2 presents how the barcodes are spaced in each version of the technology. Today, this technology is one of the most popular methods for SRT due to its accessibility and advantages regarding high throughput.

Figure 8 – Distribution of spots in ST.



Source: Adapted from (ASP; BERGENSTRÅHLE; LUNDEBERG, 2020)

In this project, the image provided for the ST dataset was HE stained. This technique provides an RGB image in which the cell nuclei are colored as blue-purple, the extracellular matrix and cytoplasm are pink, the air spaces are white, and the other components may be a combination of these colors (CHAN, 2014). Therefore, it is possible to obtain information not only about the nuclei but also of the components surrounding them. This is the most used staining method for histology; it is often called the gold standard (ROSAI, 2007). The HE image from the data collected in the laboratory had 1.8 GB in size, 30 bits per pixel, 21015 pixels in width, and 22832 pixels in height.

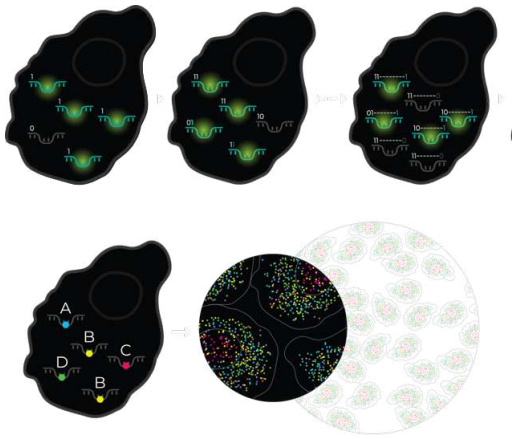
3.1.2 MERFISH

The general idea behind the FISH methods is to bind fluorescent particles to selected transcripts and then photograph this tissue to locate them using image analysis (LANGER-SAFER; LEVINE; WARD, 1982). Therefore, it is possible to obtain the exact location of the gene. However, since the transcripts are close to each other, it is not possible to detect multiple genes at once because the light emitted from one type would obfuscate the one emitted from others.

To overcome the limitations of the Fluorescence In Situ Hybridization (FISH) methods, other techniques have been developed. In this project, we used the MERFISH

method, acquired by the Vizgen company. The difference between this tool and the original FISH technique is that the MERFISH captures multiple hybridization images and then combines them using an error-correction technique (Figure 9). Therefore, it allows near-genome-wide genome profiling at subcellular resolution (MOFFITT; ZHUANG, 2016).

Figure 9 – Illustration of the transcripts in each individual cell as obtained from the MERFISH data.



Source: Vizgen

Differently from the ST, the MERFISH method does not provide the HE image of the tissue. Instead, it offers a DAPI stained image. Since it binds strongly to adenine–thymine-rich regions in DNA, it is often used to stain nuclear DNA (KAPUSCINSKI, 1995). Therefore, it does not have three color channels such as the RGB image, but just one which represents the nuclei. The DAPI image provided by Vizgen had 10.9 GB in size, and 89085x61310. Therefore, it was not possible to load the image using regular software as it would overflow the computer memory.

Additionally, the MERFISH data provides the polygon files for the cells in the image. These objects indicate the border of the cells, but instead of using a regular image mask, they present the polygon that approximates the cell shape.

3.1.3 Pseudo-ST

In this project, we aim to improve the resolution of ST datasets by segmenting the cells in the tissue and classifying their cell types using information obtained from cell-type deconvolution. However, there is no labeled data to evaluate our model. Therefore, we created a pseudo-ST dataset using the MERFISH DATA.

The MERFISH presents the matrix of genes per cell. However, if we used this information to generate the expression per spot, it would not look like the ST dataset. There are cells that can reside in the border of the spot, so a portion of its transcripts

are inside the spot and the rest is not. The ST method does not focus on extracting the information from the cell but from the spot. Therefore, it counts the genes regardless of which cell it is.

To overcome this issue, we used the matrix of gene and position to create the pseudo-ST. The first step was to generate a list with the centroid for each spot. Then, by iterating over the matrix, we verified which transcripts were within the circumference of the spot (55 μm) and added them to the expression matrix. This process is not computationally efficient, as it uses for loops, however, it was used because the creation of the pseudo-ST data would only be performed once.

3.2 IMAGE SEGMENTATION

The second step in this project was to segment the tissue image. In this section, subsection 3.2.1 describes the segmentation of the nuclei in the DAPI image, while subsection 3.2.2 explains the procedures for the HE image.

Since the image file was too large to fit in a regular computer, before actually segmenting the cells or doing any processing, we sliced it into tiles. Therefore, each tile would be analyzed individually and the results would be merged together taking into account the coordinates of the tile regarding the original image. To perform this task, we used the Terra package (HIJMANS, 2022) because it allowed cropping the image without loading it into memory.

3.2.1 DAPI Image

The DAPI image provided by the MERFISH dataset comes with polygon files representing each cell. Therefore, it could be used to train a supervised algorithm of image segmentation. However, as we aim to extend this project to the ST data, which presents an unlabeled image, we focused on unsupervised algorithms to perform the segmentation task.

In this subsection, item 3.2.1.1 explains the adjustments made in the image, while the other sections explain the segmentation methods that were used.

3.2.1.1 Image Preprocessing

Before implementing the segmentation methods, we opted to process the image to highlight the nuclei characteristics. We implemented contrast stretching (normalization), which attempts to improve the contrast by stretching the range of intensity values. We also applied histogram equalization, which adjusts the intensities by spreading out the most intense pixels and making the histogram more evenly distributed. Last, we tested the adaptive histogram equalization technique, which follows the same principles of histogram equalization but creates different histograms for the regions of the image.

3.2.1.2 Global Thresholding

Since the DAPI image has only one color channel, the simplest method to use to create masks is thresholding. This technique works by comparing each pixel of a gray image to a selected value (threshold) and then substituting it for black or white, depending on whether the condition was satisfied. The idea behind thresholding is quite simple, the question that comes with is the value to choose as the threshold.

One strategy to pick the best value for the threshold is guessing. In this case, the user can try different values and analyze which provides the best results. The problem with this strategy is that the threshold that works for an image may not work well for others. Since we had to crop the original image into tiles, we need to find the value that works for all tiles, not a single one. Another reason is that there is no standard value for the threshold. So, whenever the user analyzed a different image, it would need to guess the value again.

Another alternative for selecting the threshold value is to calculate the mean of the pixel values. This strategy is advantageous because it is simple and can be automatized. However, it may not lead to the best results

This method analyses the image histogram, iterating through all the possible threshold values, and selecting the one that minimizes the intra-class variance. The advantage of this method is that it automatically calculates the threshold and produces good results if the histogram has a well-defined bimodal distribution (deep and sharp valley between the two peaks). The disadvantage is that it may not work well outside these conditions.

3.2.1.3 Local Thresholding

The previous methods calculated the threshold value analyzing the whole image. However, from the tiles we analyzed, we realized that there is a significant difference in cell density between regions of the image. Therefore, we also considered methods that only analyze a region of the image when calculating the threshold, the local thresholding methods.

Using the Fiji Local Auto Threshold tool, we tested the nine methods available in the software: Bernsen, Contrast, Mean, Median, MidGrey, Niblack, Otsu, Phansalkar, and Sauvola. This tool also allowed adjusting the radius of the region used to calculate the local threshold (SCHINDELIN et al., 2012).

3.2.1.4 Morphological Transformations

The technique used here combined different morphological transformations. First, we defined a kernel to apply to the image. Then, we used the morphological gradient operation (the difference between dilation and erosion) to calculate the bor-

ders of the cells. With the borders, we applied the closing operation to close the holes and the opening to remove the noise.

3.2.1.5 Blob Detection

A blob can be considered an object that differs in its characteristics from the background. Since the DAPI image presents the nuclei as bright spots in a black background, this technique can be used to detect the nuclei as blobs. The simpleBlobDetector function available in the OpenCV package allows the detection of blobs using filters such as area, color, circularity, convexity, and inertia ratio.

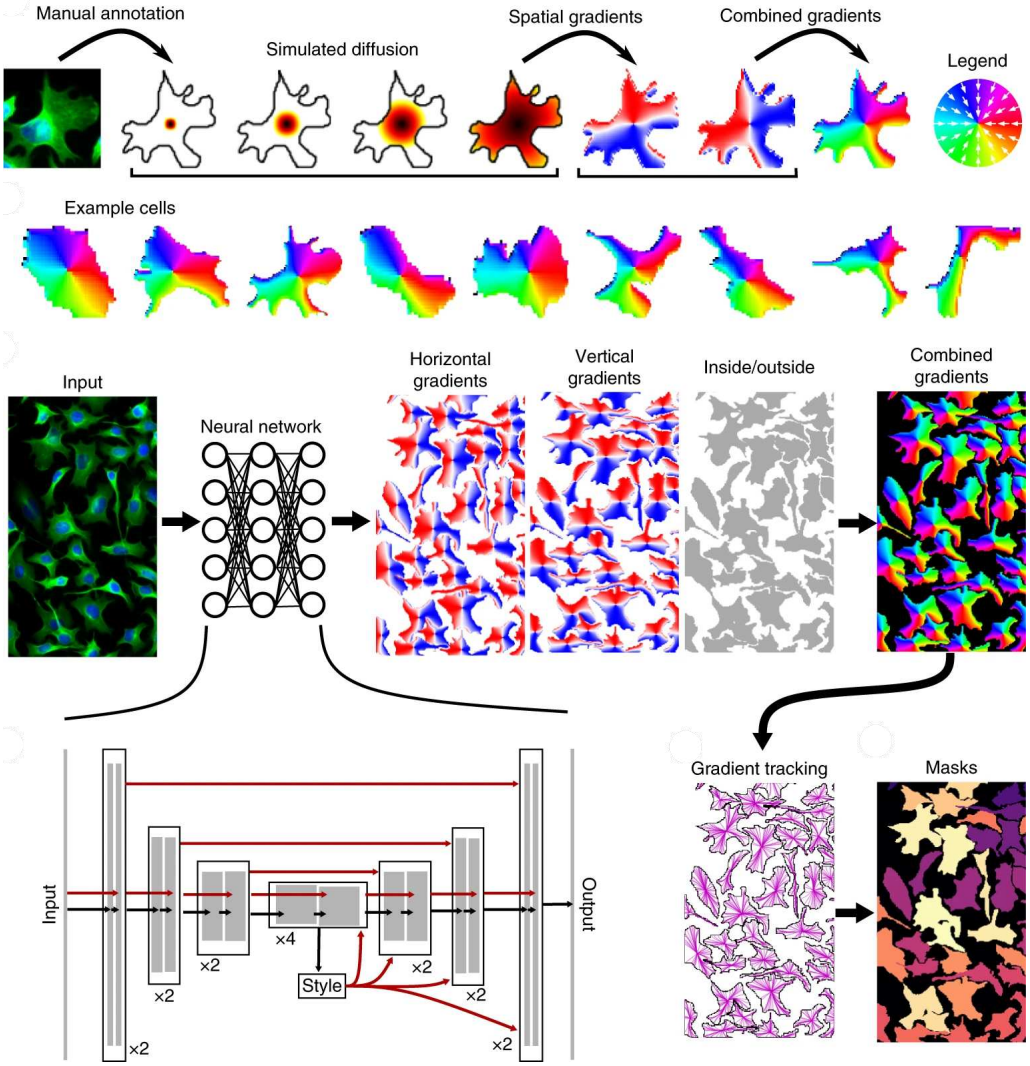
3.2.1.6 KMeans

The K-means algorithm is usually used for clustering, but it can also be used for segmentation. In this case, the method works by clustering the pixels by their intensity. Then, one of the classes can be used as the segmentation results. Similar to the K-means algorithm used for data clustering, the one for segmentation also requires the value of k to be specified by the user.

3.2.1.7 Cellpose

As there was no labeled data, we did not train our own supervised algorithm. However, we tested some pre-trained models. Cellpose is a generalist algorithm for cellular segmentation based on deep learning. It was trained on a new dataset of highly varied images of cells that contains more than 70,000 labeled data (STRINGER et al., 2021). The model combines a neural network with features extracted from the image (Figure 10).

Figure 10 – Architecture of the Cellpose algorithm.

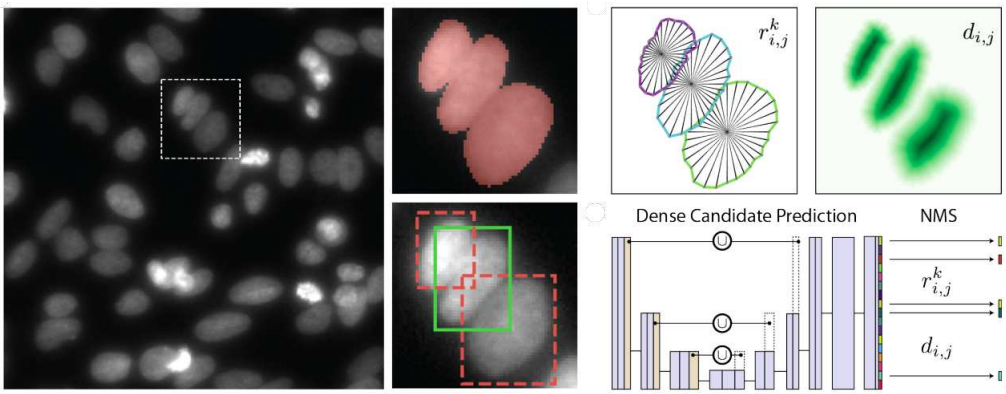


Source: Adapted from (STRINGER et al., 2021)

3.2.1.8 Stardist

Another pre-trained model tested on this project was StarDist. This algorithm proposes detecting the cells using star-convex polygons (Figure 11). It aims to overcome the limitations of previous methods that used bounding boxes to represent the cells. Its architecture is based on a convolutional neural network that predicts a polygon for the cell at the pixel position for every pixel on the image (SCHMIDT et al., 2018).

Figure 11 – Implementation of StarDist.

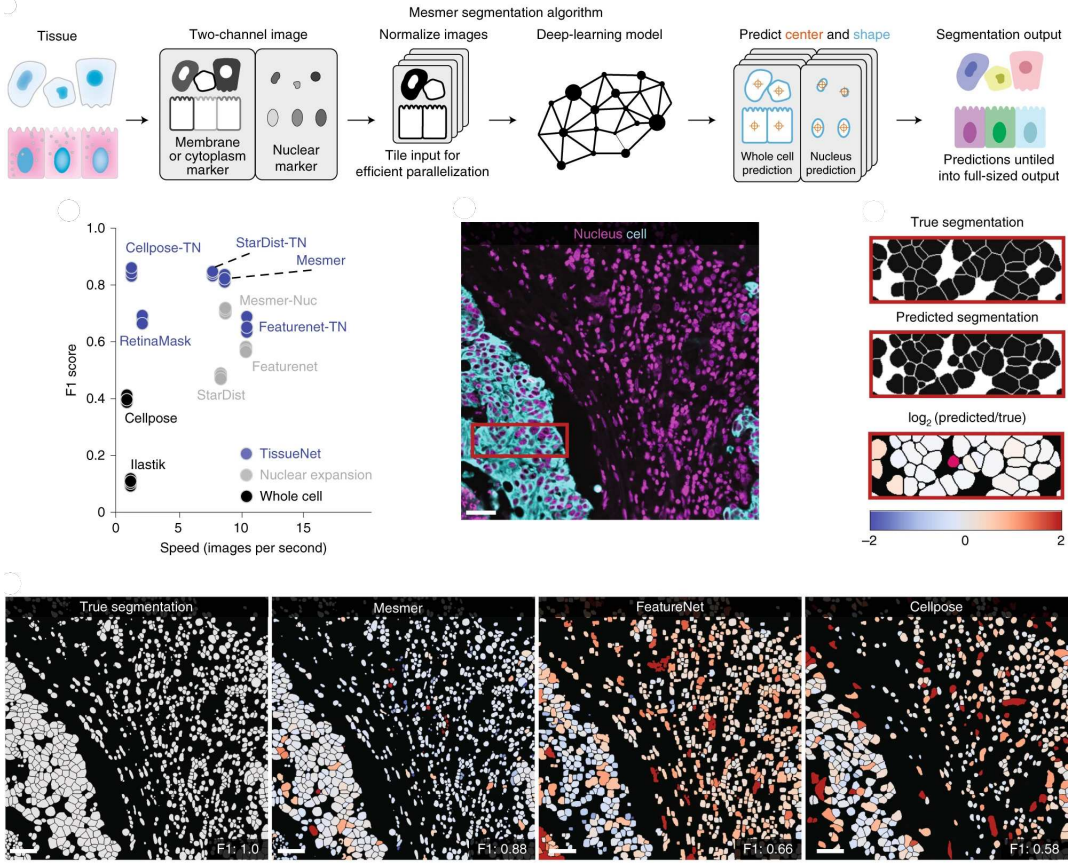


Source: Adapted from (SCHMIDT et al., 2018)

3.2.1.9 Mesmer

The Mesmer algorithm is a deep-learning-enabled segmentation model that was trained on the TissueNet dataset, which contains more than one million labeled data and uses a human-in-the-loop approach. It allows nuclear and whole-cell segmentation from different tissue images (GREENWALD et al., 2021). Figure 12 presents the architecture of the algorithm and a comparison with other models.

Figure 12 – Mesmer architecture and benchmark.



Source: Adapted from (GREENWALD et al., 2021)

3.2.2 HE Image

We focused on extending the methods applied to the DAPI image to the HE one. Since the HE provides RGB color channels, we adapted the image by extracting the nuclei channel and applying the segmentation algorithms previously described. Some of the pre-trained algorithms did not need this adaptation as they were also trained on HE images.

3.3 MORPHOLOGY INFORMATION

After segmenting the cells, the next step is to extract morphological information from them. Section 3.3.1 explains the tools studied for extracting information from the masks and section 3.3.2 from the polygons generated from the masks.

3.3.1 Masks

In this subsection, we investigated three different tools for extracting morphology information.

3.3.1.1 Fiji

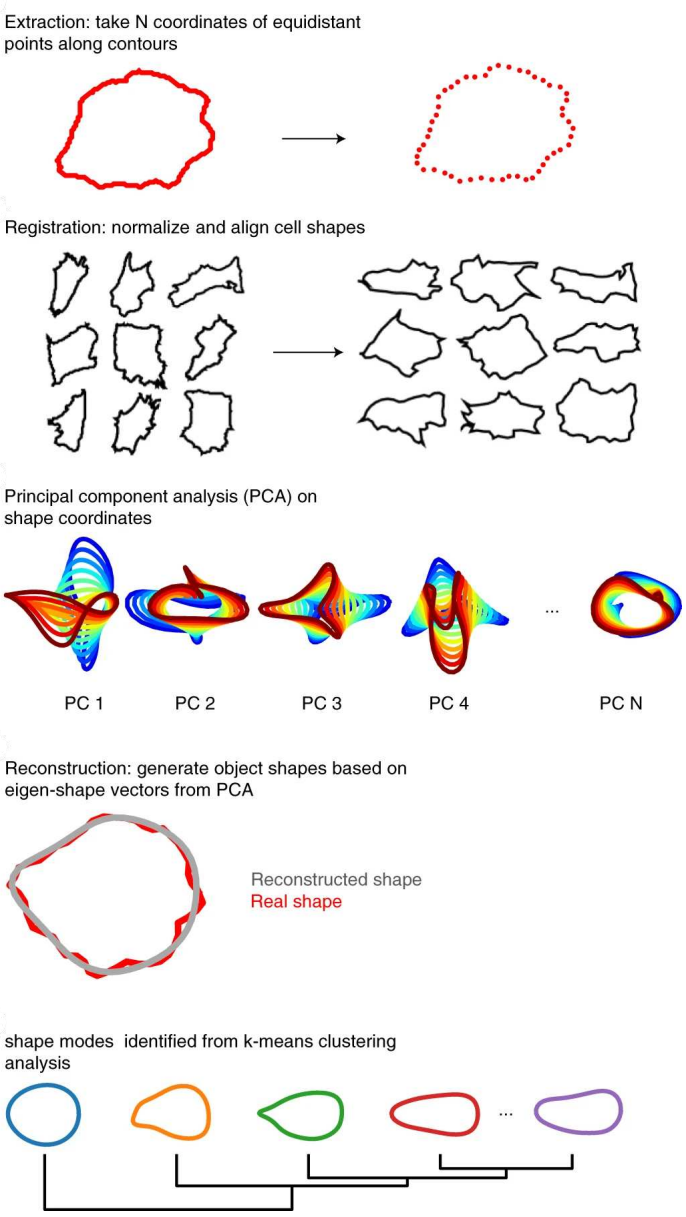
The Fiji software offers a tool called Particle Analysis. This method takes as input the binary mask and returns a matrix with the features extracted from the particles (SCHINDELIN et al., 2012). The measurements for each particle available in the platform are:

- Area
- Mean gray value
- Standard deviation of gray values
- Modal gray value
- Minimum and maximum gray values
- Centroid
- Center of mass (brightness)
- Perimeter
- Bounding rectangle
- Circularity
- Aspect ratio
- Round (roundness)
- Solidity
- Feret's diameter (maximum caliper)
- Integrated density
- Median value of the pixels
- Skewness
- Kurtosis
- Area fraction
- Stack position

3.3.1.2 Vampire

The Visually Aided Morpho-Phenotyping Image Recognition (VAMPIRE) tool enables the profiling and classification of cells obtained from post-segmentation datasets. It bases the calculation of features on equidistant points along contours. The method clusters the cells into different shape modes (PHILLIP et al., 2021). Figure (**fig:3_vampire**) presents the algorithm's pipeline.

Figure 13 – Steps in the Vampire pipeline.



Source: Adapted from (PHILLIP et al., 2021)

3.3.1.3 WND-CHARM

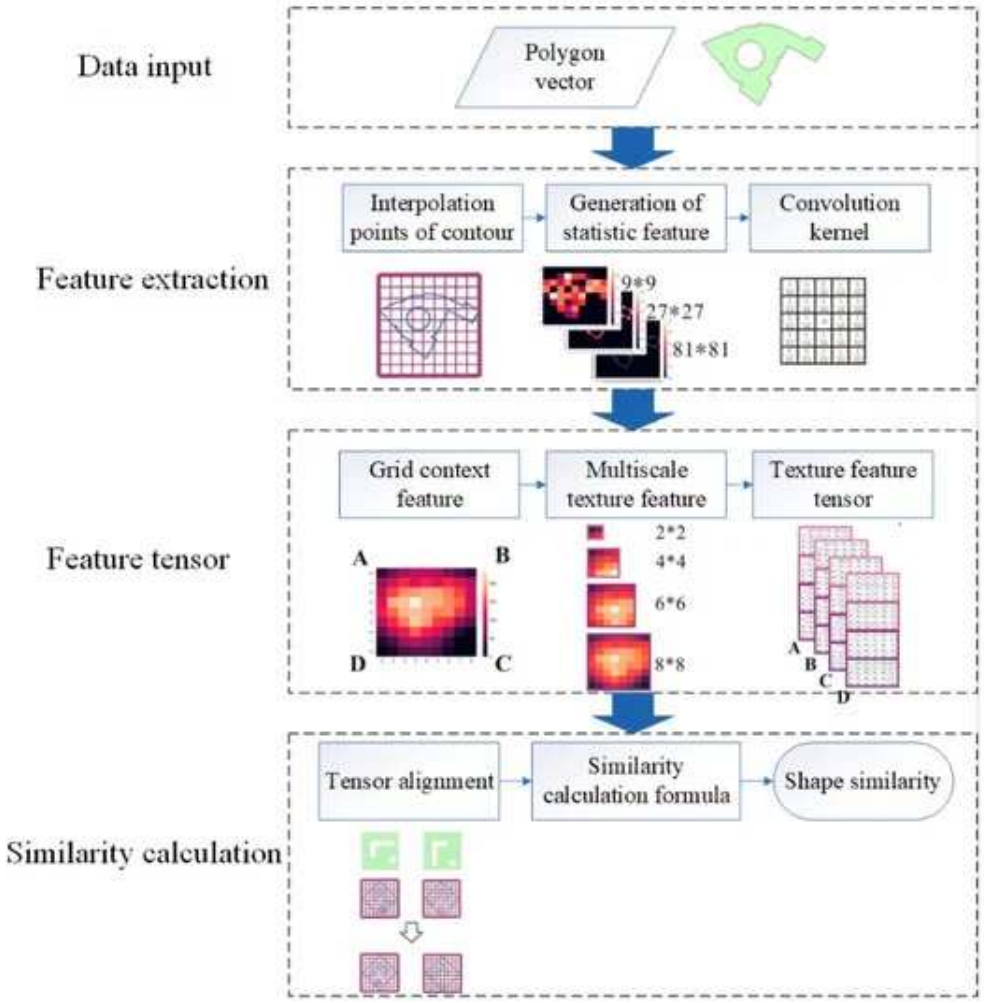
WND-CHARM is an open-source tool for biological-image analysis. It was published in 2008 and has over 3000 methods for feature extraction (ORLOV et al., 2008), including:

- Radon transform features
- Chebyshev Statistics
- Gabor Filters
- Multi-scale Histograms
- First 4 Moments, of mean, standard deviation, skewness, and kurtosis
- Tamura texture features of contrast, directionality, and coarseness
- Edge Statistics features computed on the Prewitt gradient
- Object Statistics
- Zernike features
- Haralick features
- Chebyshev-Fourier features

3.3.2 Polygons

The Giotto package from our laboratory implements a method to convert masks to polygons. We searched for tools to extract information from the polygons instead of the masks. We investigated a method that uses contour diffusion to measure the similarity of polygons (FAN; ZHAO; LI, 2021). This technique converts the polygon into a grid and applies convolutional operations to extract their characteristics (Figure 14).

Figure 14 – Overview of the method Towards Measuring Shape Similarity of Polygons Based on Multiscale Features and Grid Context Descriptors.



Source: Adapted from (FAN; ZHAO; LI, 2021)

3.4 CLASSIFICATION

Before classifying the cells, we implemented techniques to reduce the dimension of the features and visualize the cells in two dimensions. The first method used was the Principal Component Analysis (PCA) which projects the multi-dimensional data into new axes in order to maximize the variance. We also tried the t-distributed stochastic neighbor embedding (t-SNE) which works by converting the similarity of points into probabilities and then minimizes the Kullback-Leibler divergence of the low and high-dimensional space. As there were no visible clusters in the reduced dimension space, we realized that the features used were not informative and did not proceed with the classification.

4 REQUIREMENTS

This project intended to deliver a tool to enhance the resolution of spatial transcriptomic datasets by assigning the cell type obtained in the cell-type deconvolution to the specific cell location. As the final project, we aimed for an algorithm that receives the HE image, the spot locations, and the percentage of cell types in each spot; then it returns the segmented cells and their cell types. Since the ST data does not provide the ground truth for the cell types, we focused on applying the methodology to the pseudo-ST data created from the MERFISH data to later extend it to the original ST dataset. The overall objective was divided into smaller items that could be performed independently but needed to work together to reach the final goal.

The first step is to create the pseudo-ST data. The original ST dataset is composed of the image, the gene expression matrix, and the spot locations. As there is no feasible way to convert a DAPI image to an HE image, this step was discarded. The gene expression matrix should include all transcripts in the spot, regardless of whether their cell is complete inside or not. Additionally, the spot locations and sizes should match the ones generated by the 10X Visium technology (second version of the ST).

The second step is to segment the cells (or nuclei). Since the data presents a large image, the basic requirement was to process it without overflowing the memory. Then, we aimed for segmenting the cells, even those that were close to each other, and to remove the noise. Another aspect considered was the time to process the files and ease of run. Moreover, we looked for an algorithm that could be extended from the DAPI image to the HE one with few adjustments.

The third step was to extract the morphological information from the segmented cells. In this case, there is no specification of what feature should be obtained. So we focused on calculating a large number of them to later use the results to understand the characteristics that matter the most in cell-type classification. Regarding the extraction, we focused on values that were fast to calculate and easy to interpret.

After extracting the morphological information, we focused on the classification. The algorithm should produce good results (accuracy, F1-score...) when trained on the labeled data provided by the spots of pure cell type. Additionally, it needed to be interpretable in order to investigate the features that have a higher impact on the classification. It would also be useful to have a model that returns the probabilities for each cell instead of a hard answer.

With the classification ready, the following step would be to convert the results from a table to a clear visualization. Therefore, it is necessary to color the segmented cells based on their cell type and provide this information to the user. It would also be useful to highlight the border of the spots to present that some cells are on the borders.

Finally, after creating all these steps for the pseudo-ST, they should be extended

Table 1 – Functional Requirement 1.

Name: F1 pseudo-ST data		Hidden
Description: incorporate spatial data into the current package.		
Non-functional requirements		
Name	Restriction	Specifications
NF1.1 Expression matrix	Include all transcripts in the spot, regardless of the cell they belong to.	Performance; Permanent.
NF1.2 Spot shape	The size of the spots and the distance between them should be the same as those from the 10X Visium technology.	Performance; Desirable; Permanent.

Source: personal archive.

Table 2 – Functional Requirement 2.

Name: F2 segmentation		Hidden
Description: Segment the nuclei in the DAPI image.		
Non-functional requirements		
Name	Restriction	Specifications
NF2.1 Memory	Process the image without overflowing the memory.	Performance; Permanent
NF2.2 Density	Segment the cells regardless of the density of the region.	Performance; Permanent
NF2.3 Noise	Remove the noise.	Performance; Permanent
NF2.4 Speed	Be fast.	Performance; Desirable; Permanent.
NF2.5 Generalization	Be easily extendable to segment the HE image.	Performance; Desirable; Permanent.

Source: personal archive.

to the original dataset. As there is no way to evaluate the method on the ST dataset, the previous steps should be verified individually, especially the image segmentation one.

Tables 1, 2, 3, 4, 5, and 6 present an overview of all requirements. They were classified into two categories: functional items, which represented a task that needed to be completed, and non-functional items, which represented qualifications for the functional ones. They were designed to ensure that all of the procedures could work together to confirm that the biological results were usable and meaningful. Furthermore, they are utilized to determine whether the project's initial goals were met.

Table 3 – Functional Requirement 3.

Name: F3 Morphological information		Hidden
Extract the morphological information from the segmented cells.		
Non-functional requirements		
Name	Restriction	Specifications
NF3.1 Number of features	Consider a large number of features.	Performance; Permanent.
NF3.2 Speed	Be fast.	Performance; Desirable; Permanent.
NF3.3 flexibility	Be easily interpretable.	Performance; Desirable; Permanent.

Source: personal archive.

Table 4 – Functional Requirement 4.

Name: F4 Classification		Hidden
Description: Classify the cells regarding the cell type.		
Non-functional requirements		
Name	Restriction	Specifications
NF4.1 Evaluation	Produce good results even when calculated on the limited labeled data produced by the pure spots.	Performance; Permanent.
NF4.2 Speed	Be fast.	Performance; Desirable; Permanent.
NF4.3 Interpretability	Be easily interpretable.	Performance; Desirable; Permanent.
NF4.4 Results	Return the probabilities instead of only the classes.	Performance; Desirable; Permanent.

Source: personal archive.

Table 5 – Functional Requirement 5.

Name: F5 Visualization		Hidden
Description: Visualize the results.		
Non-functional requirements		
Name	Restriction	Specifications
NF5.1 Images	Present the classified cells in their original image.	Performance; Permanent.
NF5.2 Spots	Highlight the border of the spots.	Performance; Permanent.

Source: personal archive.

Table 6 – Functional Requirement 6.

Name: F6 Generalization		Hidden
Description: Apply the methodology to the ST dataset.		
Non-functional requirements		
Name	Restriction	Specifications
NF6.1 Verification	Verify each requirement individually.	Performance; Permanent.

Source: personal archive.

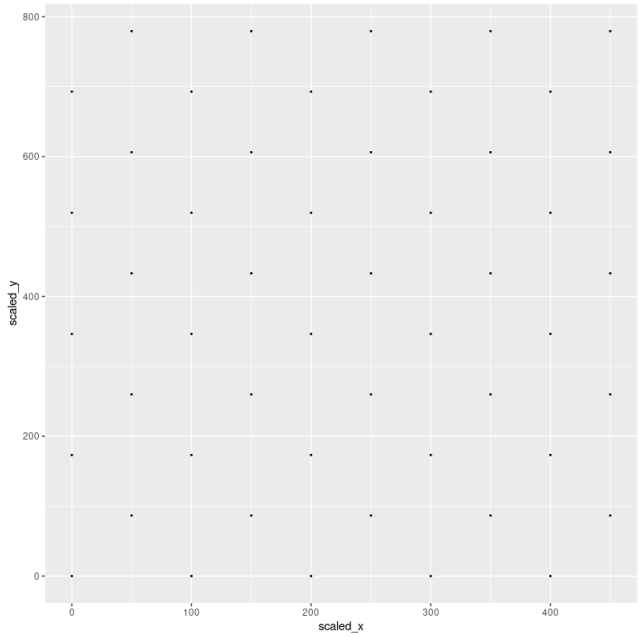
5 DEVELOPMENT

In this chapter, we present the actual development of the project and the results obtained. Section 5.1 describes the creation of the pseudo-ST data. Section 5.2 the image processing and segmentation. Section 5.3 describes the efforts to extract morphological information. Finally, section 5.4 shows the steps for the classification.

5.1 PSEUDO-ST

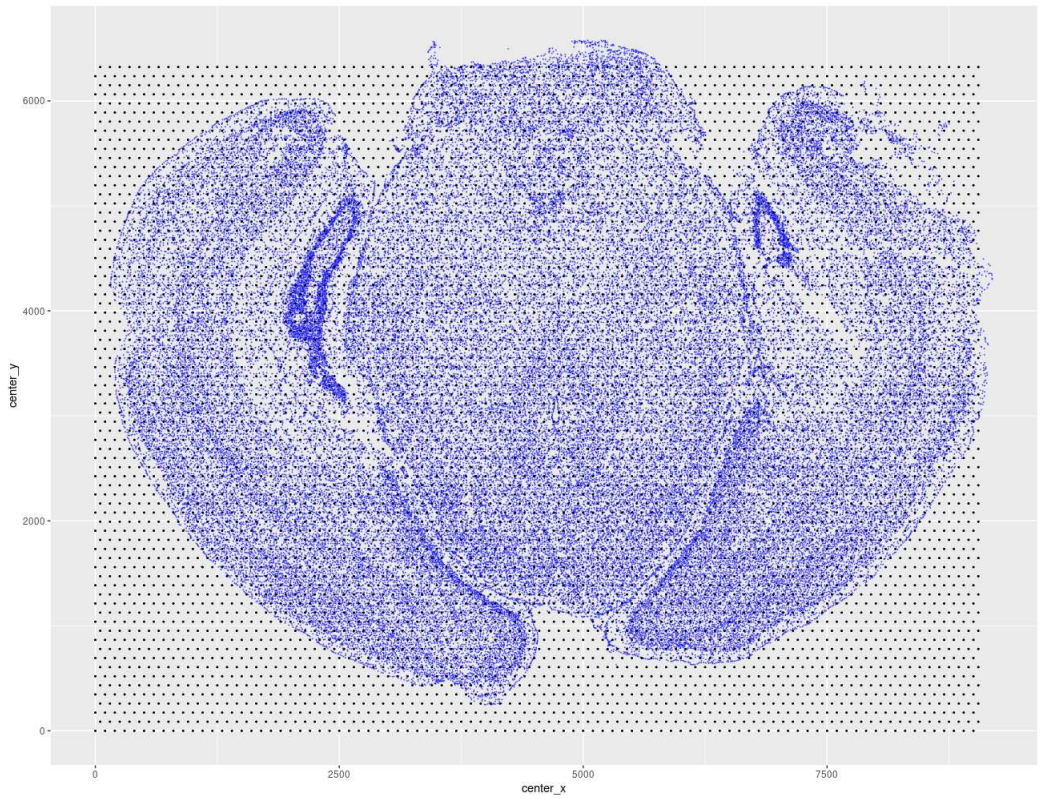
To generate the pseudo-ST dataset, we first created a list with the centroids for the spots. Following the characteristics of the 10X Visum data, we aimed to space them 100 μm apart from each other. Therefore, we create a row with the spots separated by this distance. Then, created another row with the centers shifted by 50 μm and 86.6 μm below the other row, creating an equilateral triangle with the vertex as the centers. Figure 15 shows a section of the centroids and Figure 16 presents the center of the centroids over the cells in the MERFISH dataset.

Figure 15 – Center of sample spots.



Source: Personal archive.

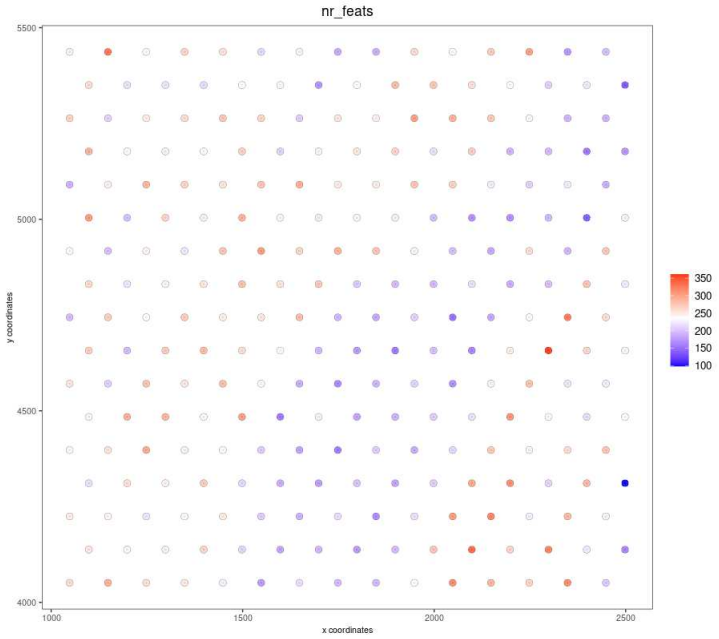
Figure 16 – Spots over the cells of the MERFISH data.



Source: Personal archive.

After calculating the centroids, we counted the transcripts within the spot region. We used for loops to iterate over the spots, and if statements to verify if the euclidean distance of the transcript to the center was smaller than $50\mu m$. To optimize the process, we filtered the transcripts based on their coordinates before calculating the euclidean distance. We also opted to select only a piece of the complete image, since it would be faster to test and require less memory. Figure 17 shows the number of transcripts in the selected section.

Figure 17 – Number of transcripts in each spot.

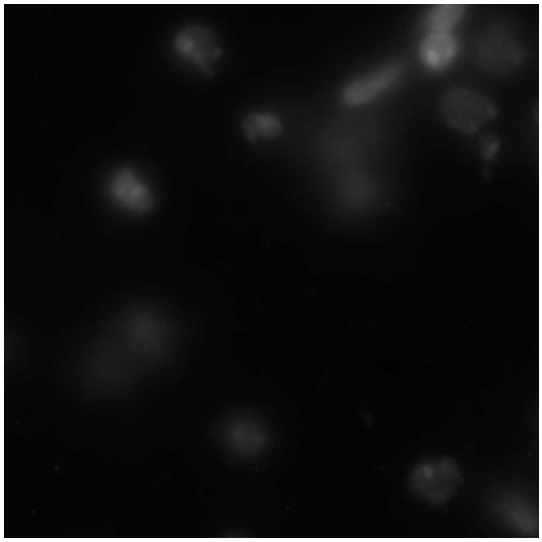


Source: Personal archive.

5.2 SEGMENTATION

In this section, we describe the preprocessing steps evaluated in the project in subsection 5.2.1 and we show the methods analyzed for image segmentation in the other subsections. In all these cases, the image used as an example was the one presented in Figure 18.

Figure 18 – Example slice used in the segmentation methods.

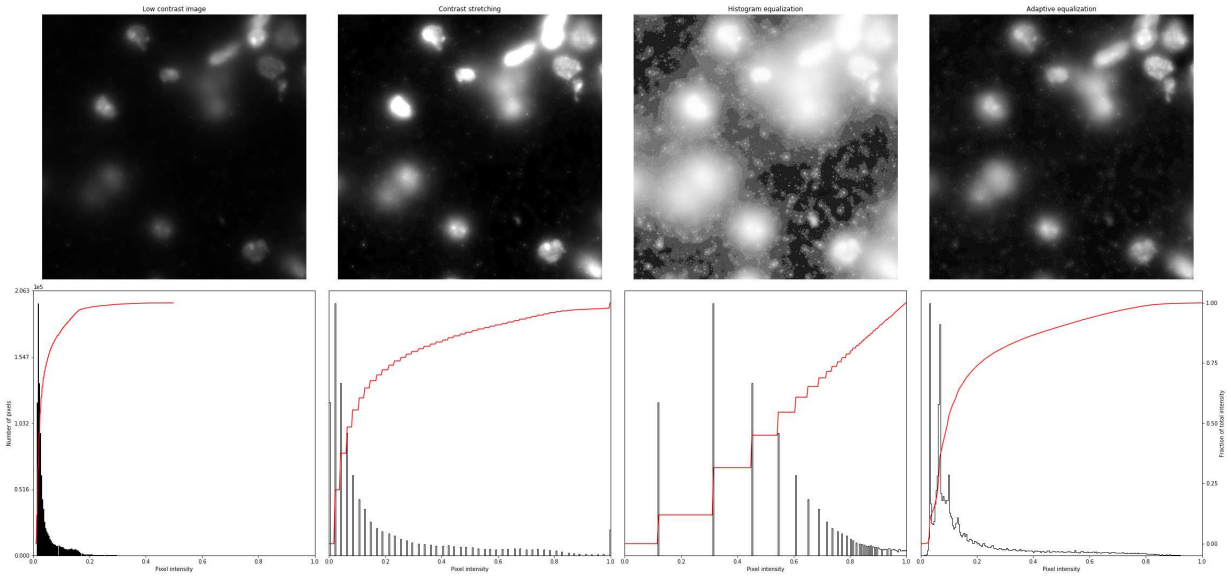


Source: Personal archive.

5.2.1 Preprocessing

By analyzing the DAPI image, we observed that it could benefit from more contrast when performing the segmentation. Therefore, we applied different contrasting techniques to observe the ones that produced the best results. Figure 19 shows the results of different histogram equalization techniques.

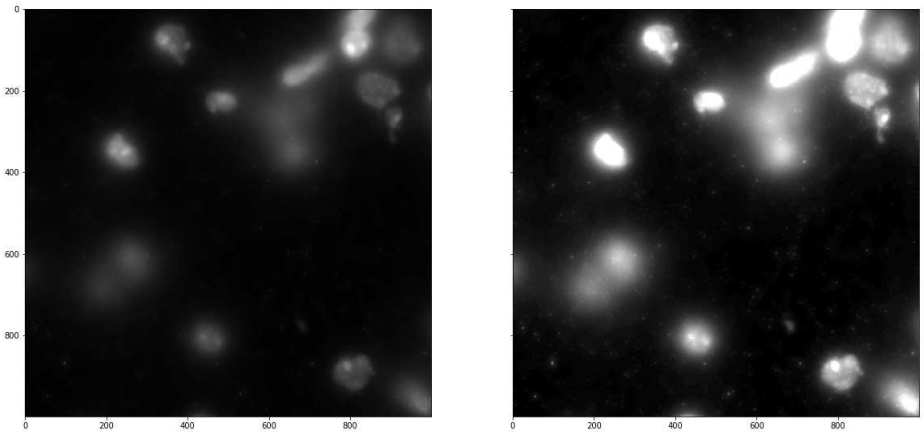
Figure 19 – Histogram equalization techniques on the DAPI image.



Source: Personal archive.

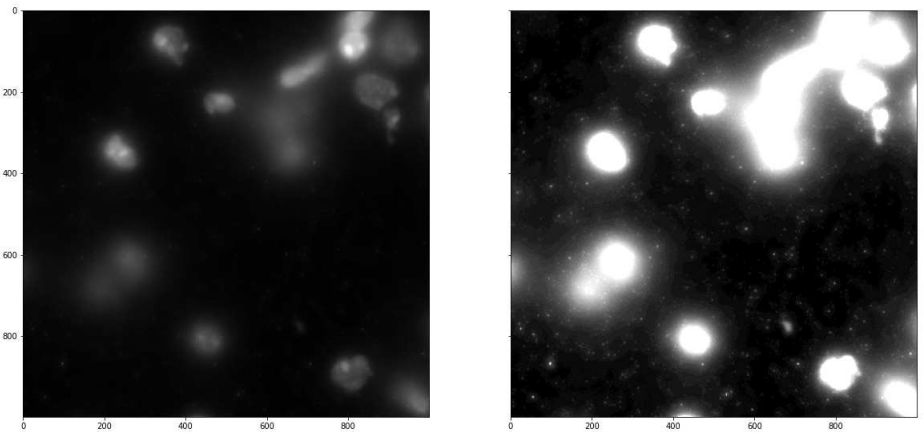
In all the cases, the processing highlighted the nuclei, but it also brought more noise. Furthermore, in some cases, it merged cells that were close to each other, making it impossible to detect them individually. Figure 20 presents a comparison between the original image (left) and the one obtained by the contrast stretching method. It is possible to see that the nuclei in the top right corner were joined into a single object after the stretching. We obtained more inadequate results when reducing the interval for the stretch (Figure 21). Therefore, we only used the original image in the segmentation.

Figure 20 – Results from the contrast stretching method with the limits of 2% and 98%.



Source: Personal archive.

Figure 21 – Results from the contrast stretching method with the limits of 10% and 90%.



Source: Personal archive.

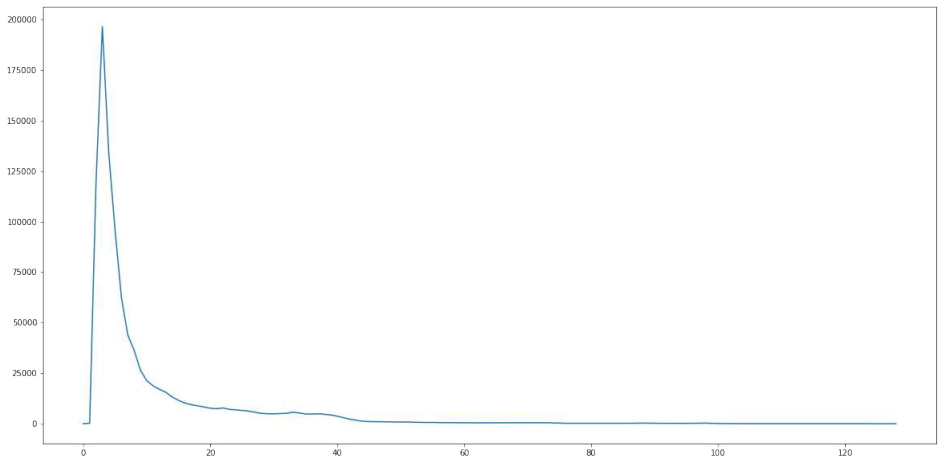
5.2.2 Thresholding

In this subsection, we present the results for all the thresholding methods tested: guessing, mean, and Otsu.

5.2.2.1 Guessing

Before actually segmenting the nuclei, we analyzed the image histogram to get an idea of the best threshold value (Figure 22).

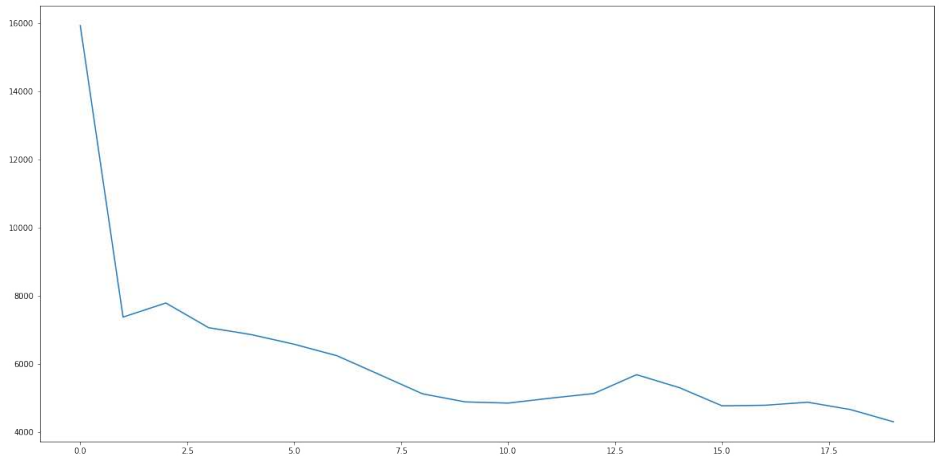
Figure 22 – Histogram of the DAPI image slice.



Source: Personal archive.

By analyzing the histogram, we noticed that there are many pixels with low intensity, probably indicating the background, and also a significant amount with intensity between 20 and 41. Therefore, we created another histogram to investigate only this region (Figure 23).

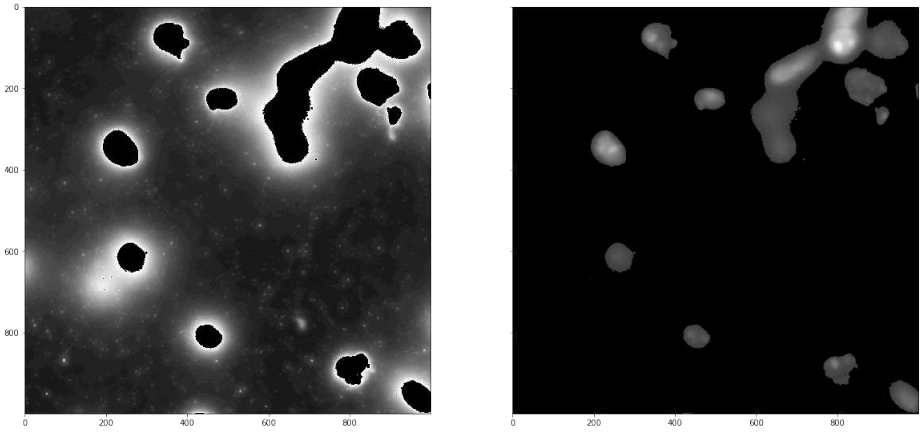
Figure 23 – Histogram of the DAPI image slice for the intensities between 20 and 41.



Source: Personal archive.

Analyzing the graph in Figure 23, we chose the value 30 as our threshold because it was in one of the valleys and therefore could separate the actual nuclei pixels from the background. Figure 24 presents the results from this threshold, on the left are the pixels that were below the value, and on the right are those that were above, indicating the mask. This threshold was able to separate some of the nuclei, but it failed when they were close to each other, such as for those on the top right corner.

Figure 24 – Thresholding by guessing the value of 30 based on the image histogram.



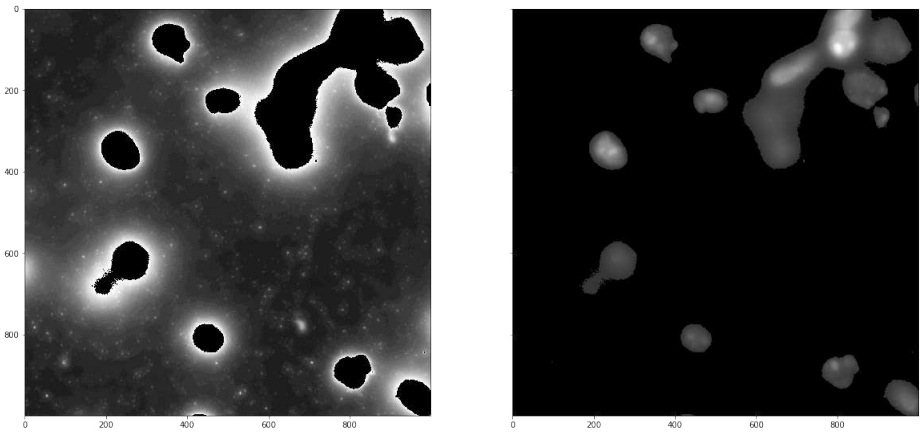
Source: Personal archive.

5.2.2.2 Mean

Next, we calculated the mean intensity of the pixels and used it as the threshold. As this slice of the image does not have many nuclei and most of it is background, then the mean value was 11, less than half of what we guessed from the histogram.

Figure 25 shows that this technique also did not present good results as the nuclei were merged.

Figure 25 – Thresholding by the mean value of 11.

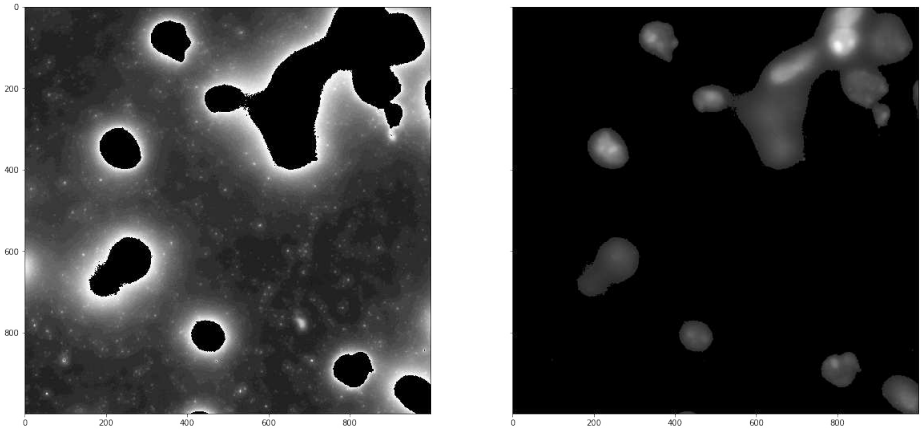


Source: Personal archive.

5.2.2.3 Otsu's Method

We applied Otsu's thresholding method to the image and found the value of 22, a little less than our guess. The results produced by this value are shown in Figure 26. As the previous results, the algorithm did not work well for dense regions.

Figure 26 – Thresholding by Otsu's method with the value of 22.



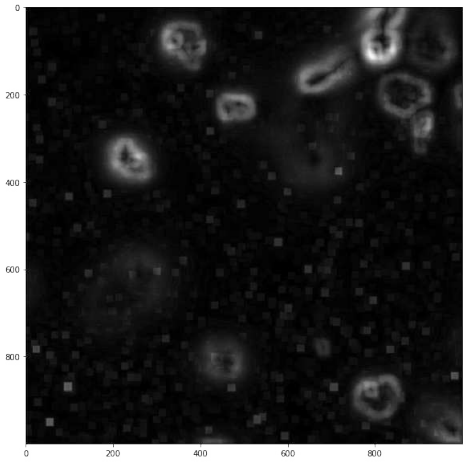
Source: Personal archive.

5.2.3 Morphological Transformations

We tried to segment the nuclei by applying the image gradient followed by morphological operations. Using a kernel of ones with size 5x5 pixels, we calculated the

gradient of the image (Figure 27).

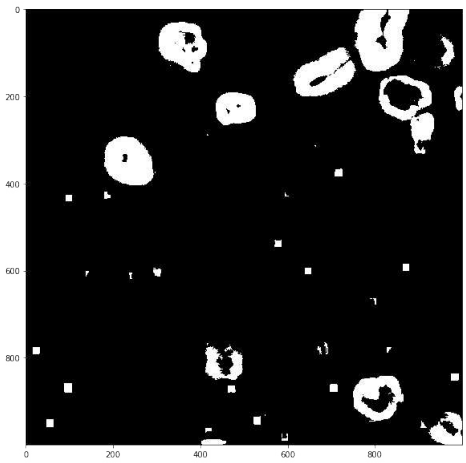
Figure 27 – Gradient with a kernel of ones with the size of 5x5 pixels.



Source: Personal archive.

We applied the OR operation on the inverse binary with the Otsu thresholding. Then, we inverted the image and obtained the result of Figure 28.

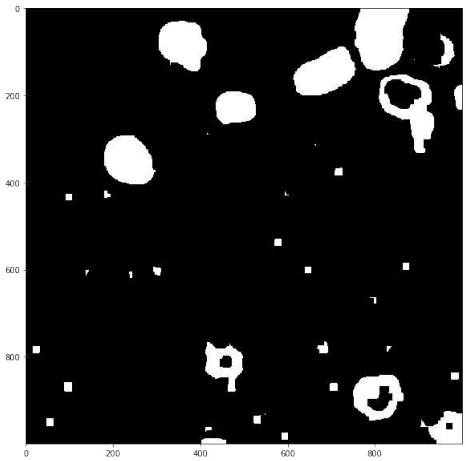
Figure 28 – Inverted results of the bitwise or operation between the inverted binary and Otsu thresholds.



Source: Personal archive.

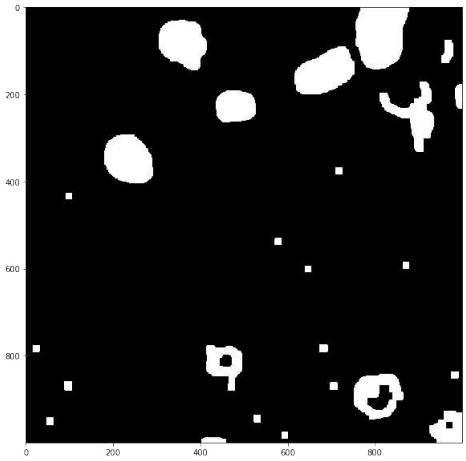
We applied the closing operation to close the holes in the nuclei (Figure 29). Then, we used the opening operation to remove the noise (Figure 30).

Figure 29 – Results of the closing operation.



Source: Personal archive.

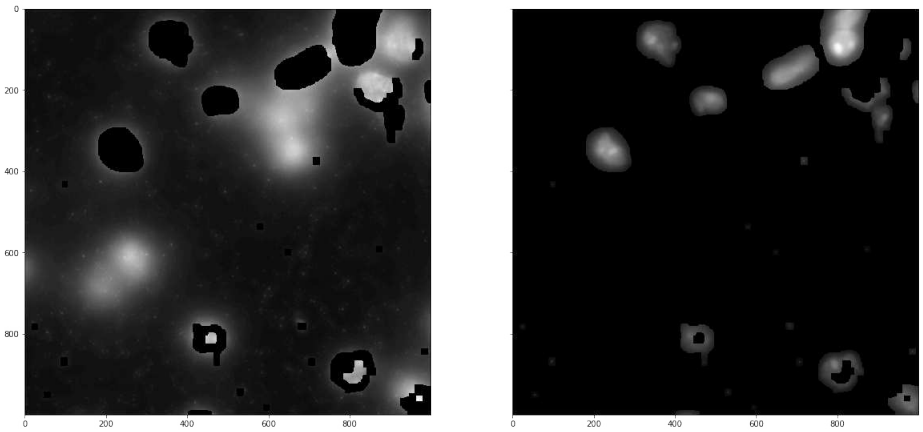
Figure 30 – Results of the opening operation.



Source: Personal archive.

From the previous images, it can be seen that the nuclei were not perfectly segmented and that there is still some noise in the image. Figure 31 presents the mask for these operations and confirms that the method was not successful.

Figure 31 – Results from the morphological operations.

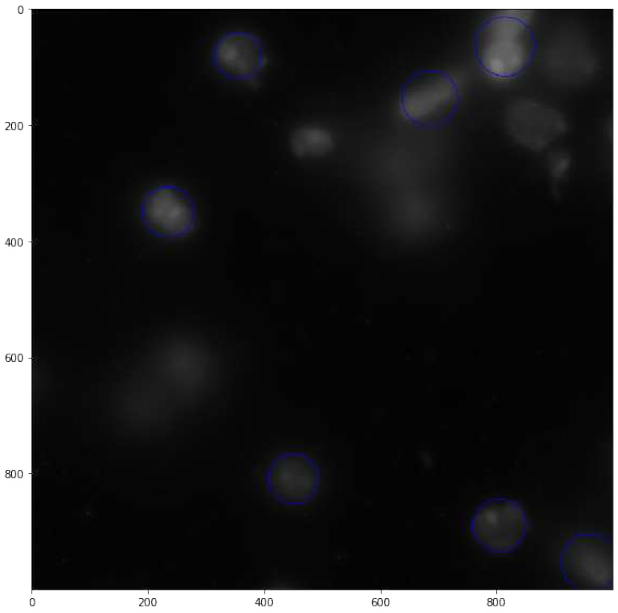


Source: Personal archive.

5.2.4 Blob Detection

For the blob detection algorithm, we did not choose to filter by circularity, convexity, or inertia ratio. Instead, we filter the blobs by color, indicating that they were the bright spots, and by area, that should be between 2500 and 12500 pixels. These values were obtained after testing different combinations. Figure 32 presents the 7 blobs detected in the image and shows how the algorithm failed to detect most of the nuclei.

Figure 32 – Blobs detected in the image.

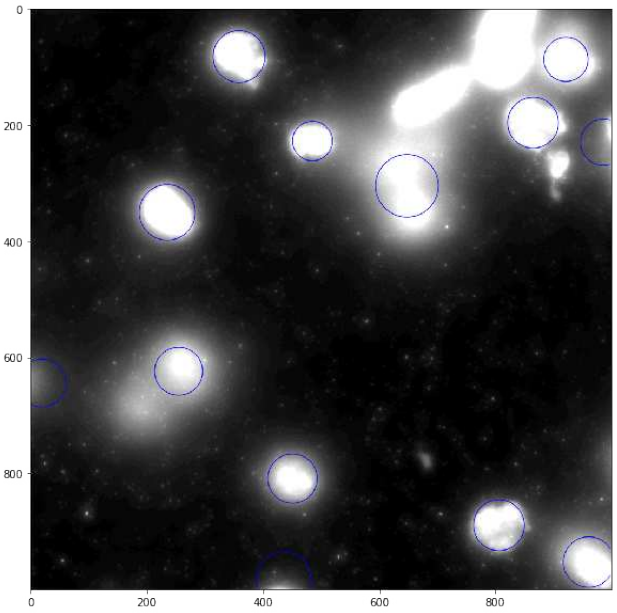


Source: Personal archive.

In an attempt to improve these results, we used the algorithm on the image with

adjusted contrast. Figure 33 shows that the algorithm detected 13 blobs, but still failed in some of the cases.

Figure 33 – Blobs detected in the contrast-adjusted image.

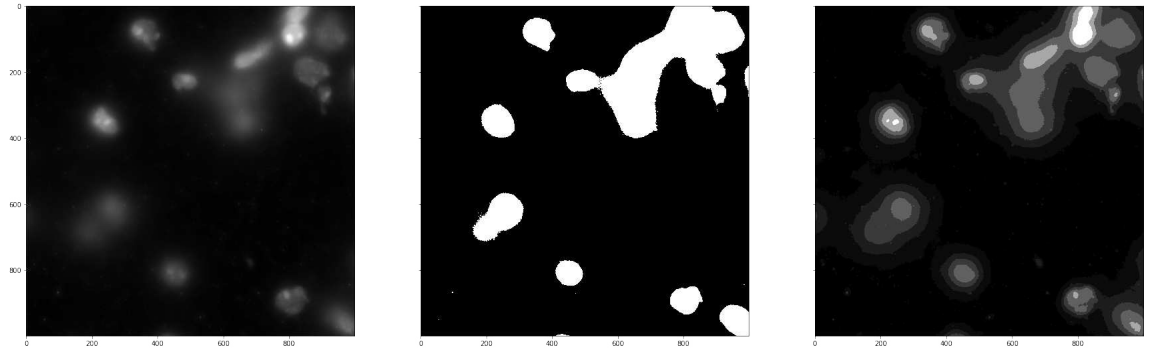


Source: Personal archive.

5.2.5 K-means

We used the K-means algorithm to segment the image by creating clusters based on the pixel intensity. Figure 37 shows the results of the clustering with k equals 2 (middle image) and 7 (right image). From the image with 2 clusters, we observe that some of the nuclei were merged into a single blob, similar to what happened in the thresholding. For the image with 7 clusters, it is possible to observe the nuclei more clearly, but to convert these clusters into actual masks, we would need to process the image. Furthermore, even with k equals 7, some of the nuclei were merged.

Figure 34 – Original image, results from the K-means algorithm for k equals 2 and 7, respectively.

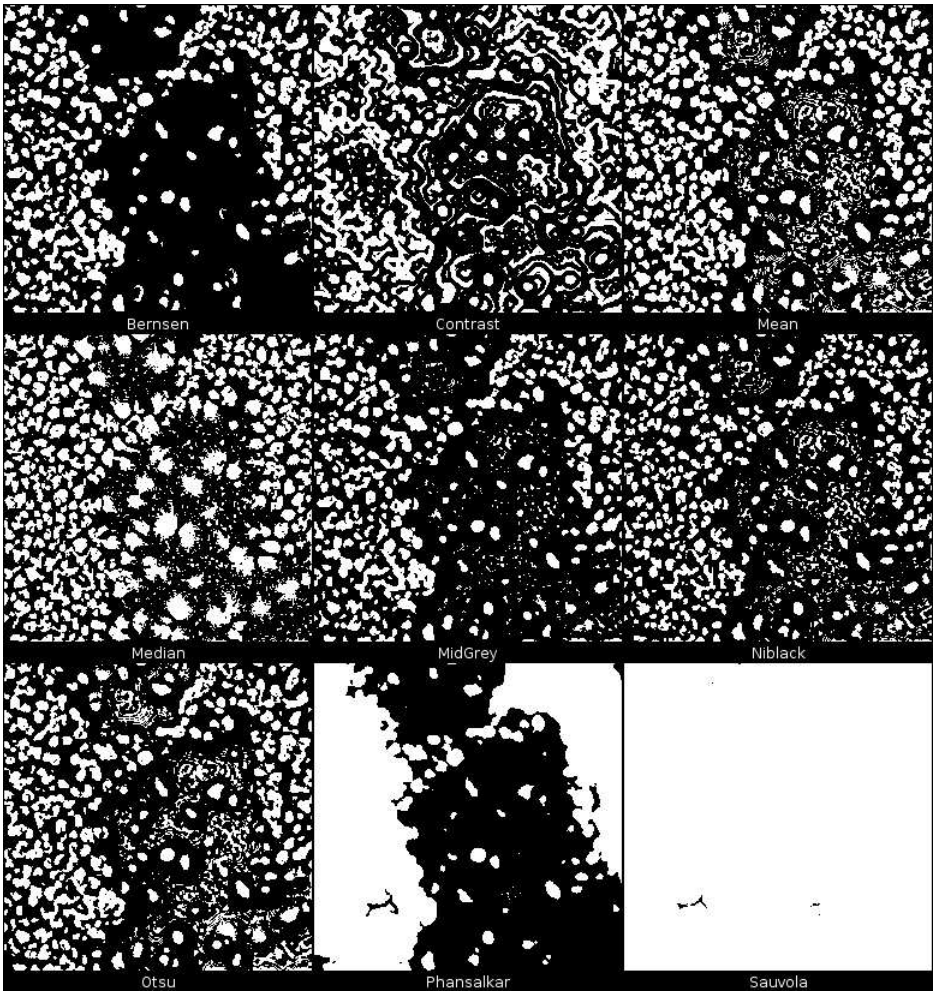


Source: Personal archive.

5.2.6 Local Thresholding

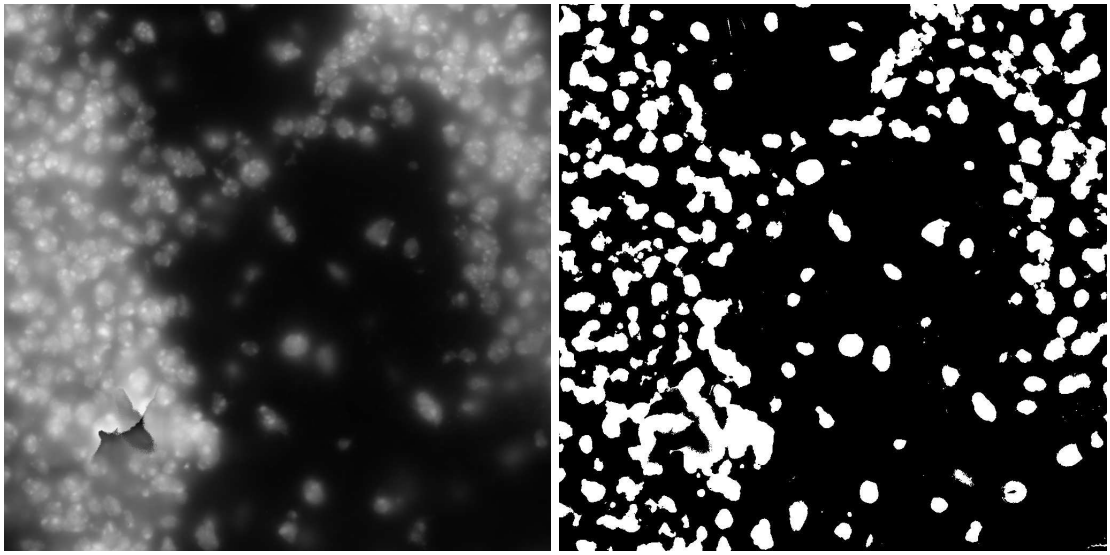
After realizing that most algorithms did not perform well on the denser regions of the image, we investigated local thresholding methods since they only analyze a part of the picture to calculate the threshold. Therefore, areas with fewer cells could have a smaller value while areas with more cells could have a greater one. The Fiji software already provides an automated local threshold with different methods. Figure 35 shows the results for the segmentation considering an area of 60 pixels for local thresholding. The Bernsen (Figure 37) and Midgray (Figure 37) methods presented promising results that when applied the morphological operations to close the holes and remove the noise could look very similar to the actual nuclei. Choosing a smaller value for the area can lead to more noise and a greater value can aggregate more cells

Figure 35 – Auto local threshold with an area of 60 pixels for different methods.



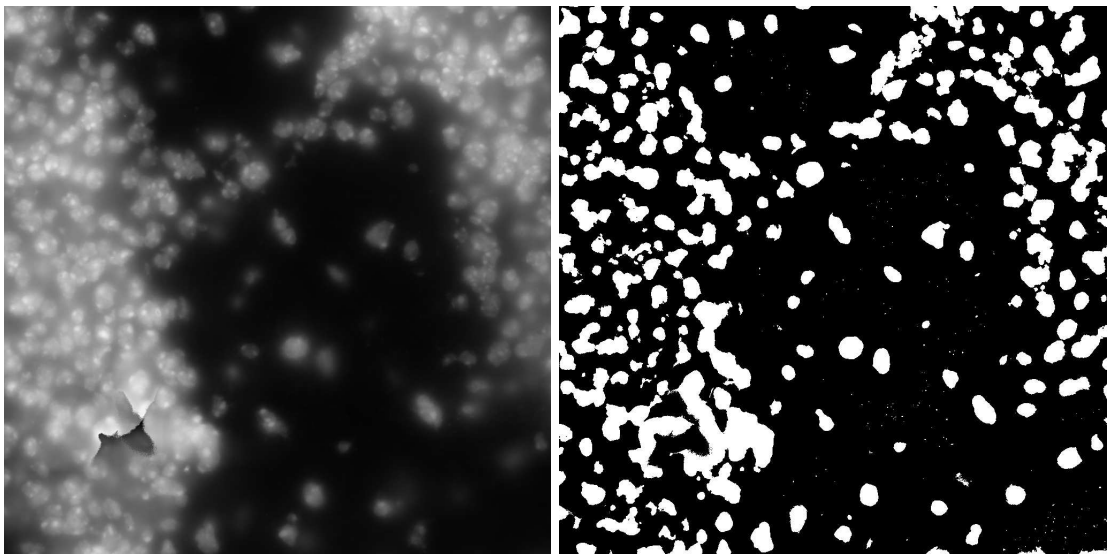
Source: Personal archive.

Figure 36 – Bernsen local threshold with an area of 60 pixels.



Source: Personal archive.

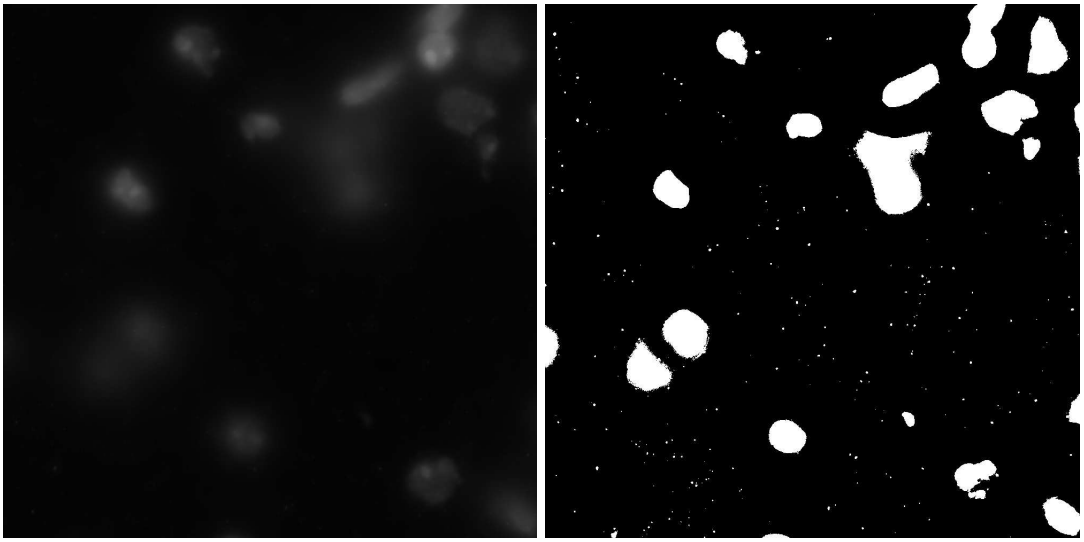
Figure 37 – MidGray local threshold with an area of 60 pixels.



Source: Personal archive.

We also applied the Bernsen method to another tile of the image, keeping the same parameters. Figure 38 shows that, in this case, the method did not provide satisfactory results as it still merged some of the cells.

Figure 38 – Bernsen local threshold with an area of 60 pixels.

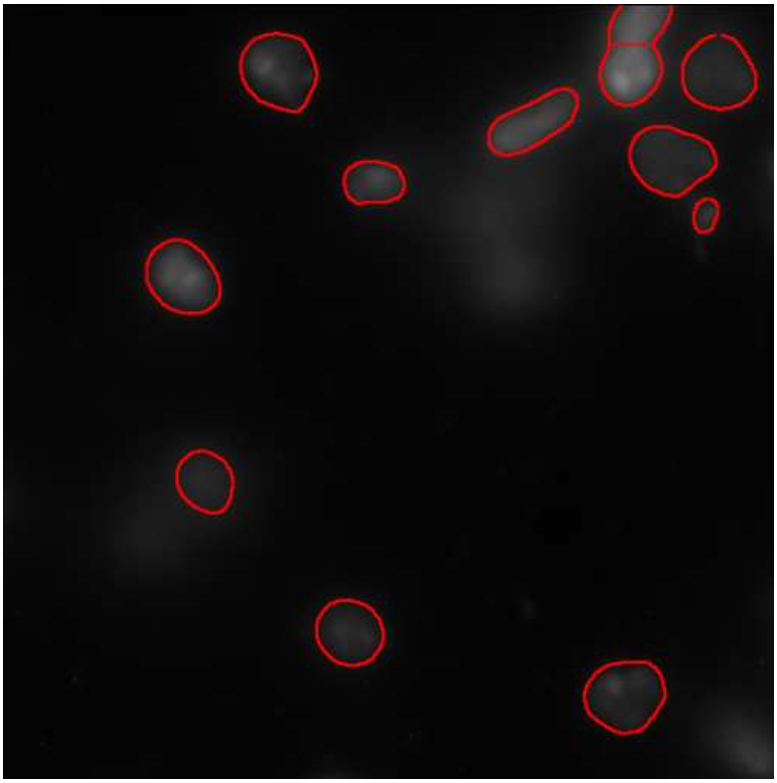


Source: Personal archive.

5.2.7 Cellpose

The CellPose algorithm takes as an input the gray image corresponding to the nuclei and the average size of the nuclei diameter. In this case, after observing and testing different values, we defined the average nuclei diameter as 50 pixels. Figure 39 presents the segmented image for this diameter. The CellPose presented better results than the previous algorithms as it was able to segment cells that were next to each other on the top right corner, but it failed for other cells, probably because they were bigger than the average diameter size.

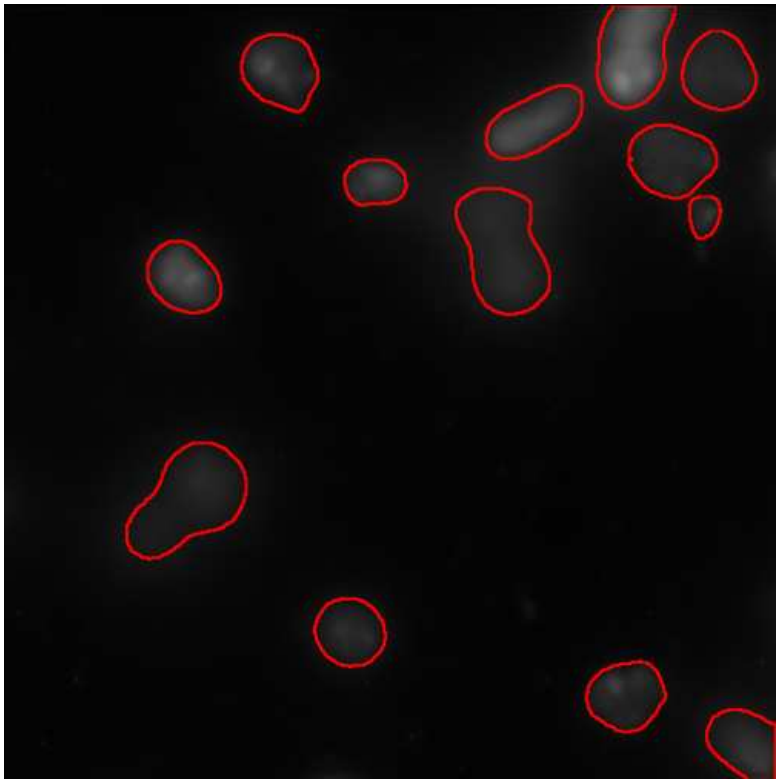
Figure 39 – Results from the Cellpose algorithm with an average cell diameter of 50.



Source: Personal archive.

Figure 40 exemplifies what happens if we increase the diameter to detect the larger nuclei. We see that the cells that were not detected previously are now identified, but the code combined nuclei that were separated before.

Figure 40 – Results from the Cellpose algorithm with an average cell diameter of 70.

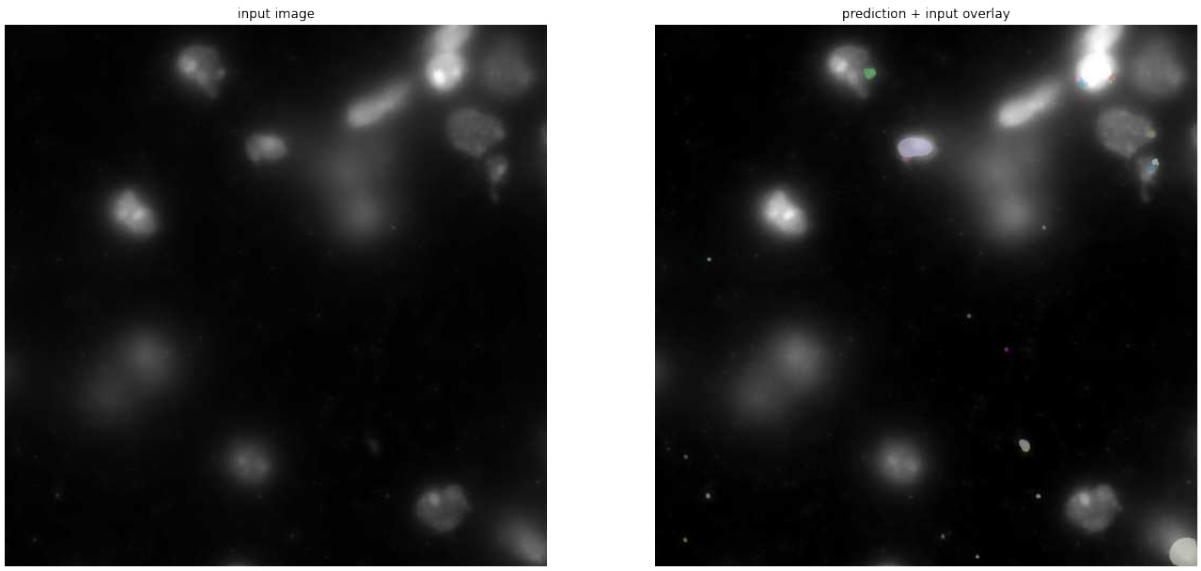


Source: Personal archive.

5.2.8 StarDist

The StarDist Python package presents a collection of pre-trained models. First, we tested the 2D_paper_dsb2018 one. Figure 41 shows that the algorithm failed to detect most of the nuclei.

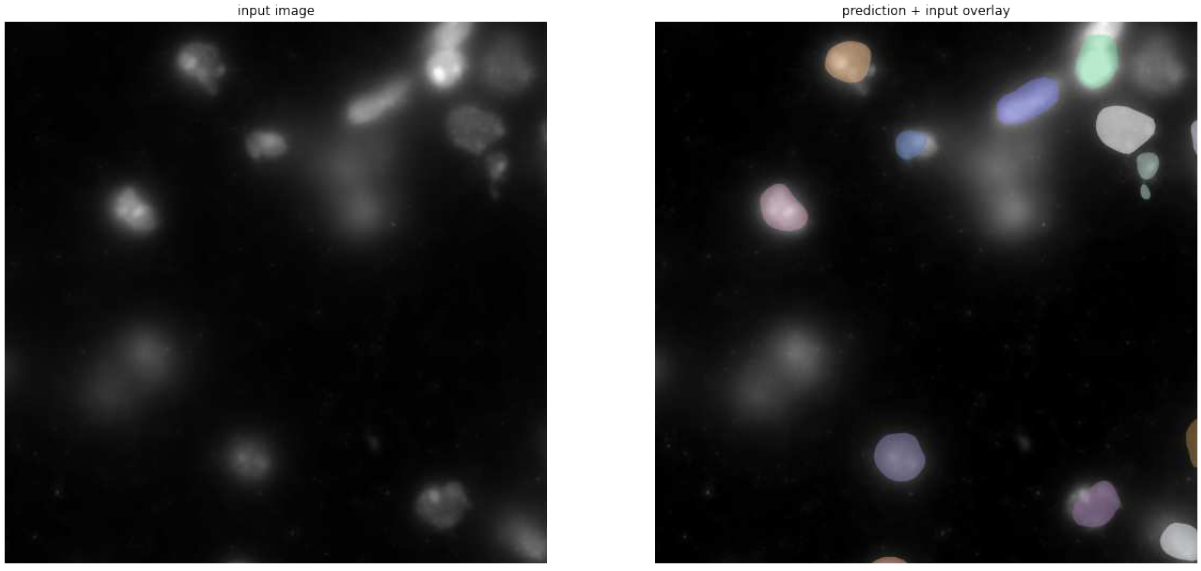
Figure 41 – Results of the 2D_paper_dsb2018 StarDist model.



Source: Personal archive.

As the 2D_paper_dsb2018 did not provide satisfactory results, we tested the 2D_versatile_fluo model. Figure 42 shows how it detected some of the nuclei, but missed or merged others.

Figure 42 – Results of the 2D_versatile_fluo StarDist model.

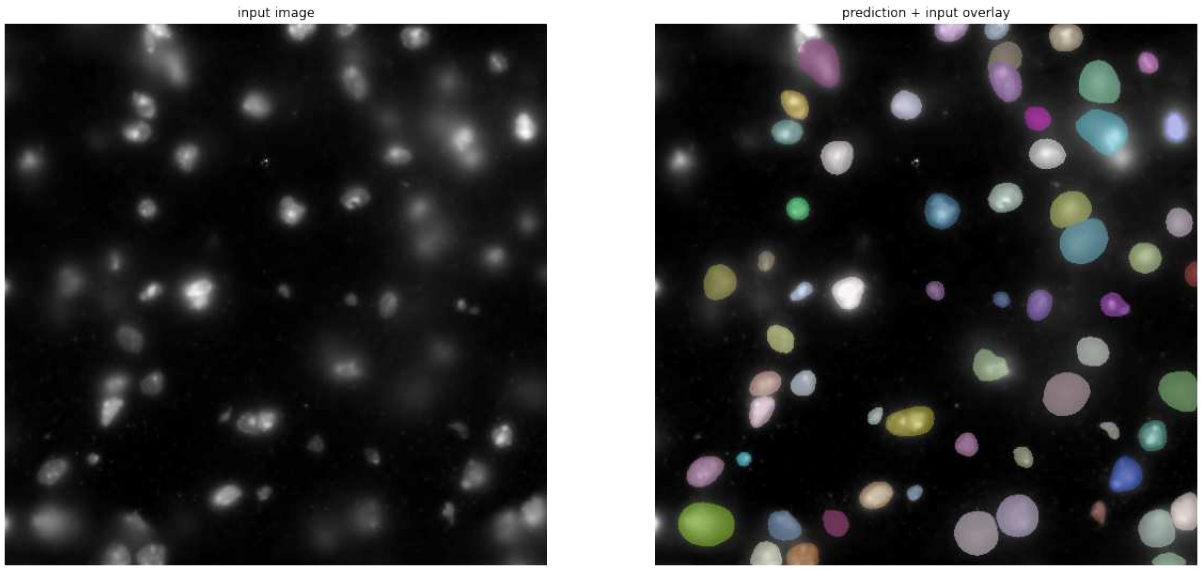


Source: Personal archive.

As there is no parameter to adjust in the StarDist model, such as the average diameter for the cell, we reduce the resolution of the image to understand how it would

impact the results. Figure 43 shows that the algorithm performed well for most of the cases, but still missed a few nuclei.

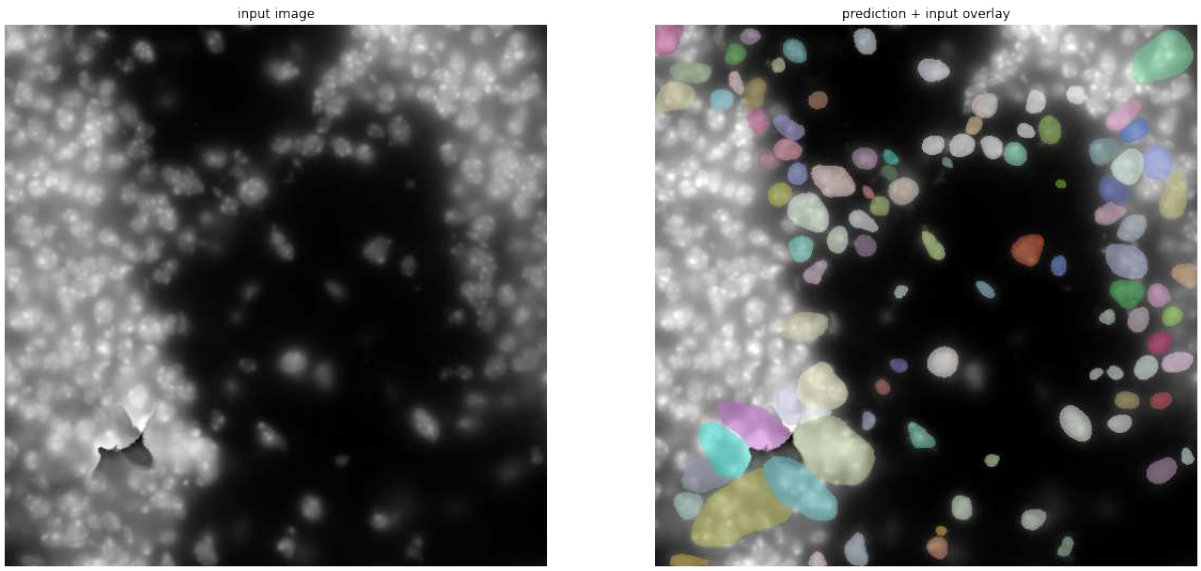
Figure 43 – Results of the 2D_versatile_fluo StarDist model with an image of reduced dimension.



Source: Personal archive.

To verify the extent to which the StarDist model produces good results, we tested it on a denser region on the image. Figure 44 shows that when there are more cells close to each other, the algorithm does not perform well.

Figure 44 – Results of the 2D_versatile_fluo StarDist model for a denser region.

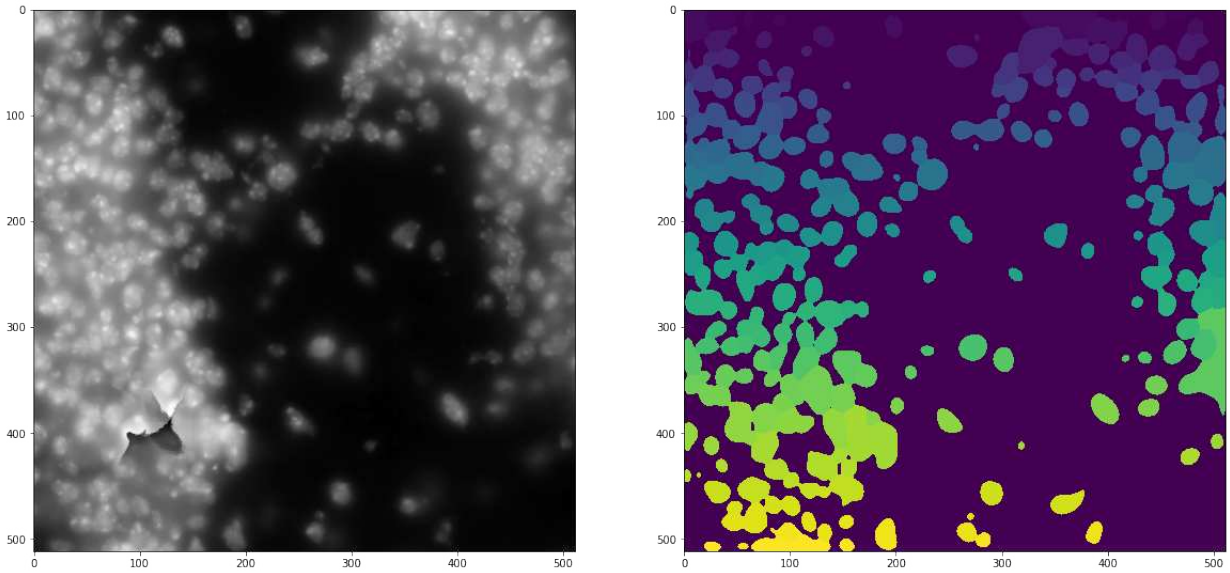


Source: Personal archive.

5.2.9 Mesmer

The last algorithm we tested was Mesmer. This model takes as input a channel for the nuclei, one channel for the cytoplasm, and the correspondence of microns per pixel. Since the DAPI image does not provide the cytoplasm channel, we passed the nuclei as this argument, following the recommendations of the paper (GREENWALD et al., 2021). Additionally, there is no adjustment on the cell size, only on the microns per pixel. In the project, we opted to reduce the resolution of the image four times and keep the value of 0.5 microns per pixel, which was the one that the algorithm was originally trained for. Therefore, we also reduced the number of tiles that the model would need to analyze and increase the speed of the segmentation. The right image in Figure 45 shows the masks for the segmentation, where each color represents a cell.

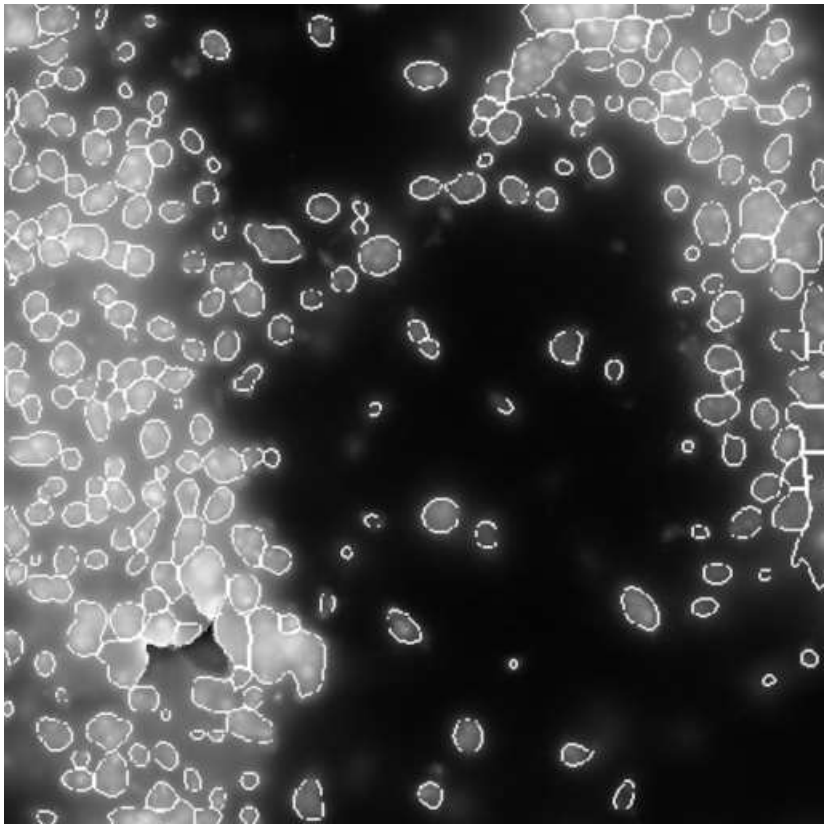
Figure 45 – Results from the Mesmer algorithm.



Source: Personal archive.

Figure 46 shows the contours of the segmentation. It is possible to see how the model performed well even in a region with cells close to each other and outperformed the previous ones. However, it still missed a few spots. Additionally, it segmented some smaller parts, which are only noise and not cells, and must be removed.

Figure 46 – Contours obtained from the Mesmer algorithm.



Source: Personal archive.

The Cellpose, StarDist, and Mesmer algorithms performed similarly when trained on the TissueNet data and evaluated using the F1-score (GREENWALD et al., 2021). However, using only the pre-trained models, the Mesmer algorithm provided the best results compared to the other methods and we chose it as the core for the segmentation. Then, we created a wrapper function to take a large image, and apply a sliding window for the segmentation. It uses the R language and the Terra package, to receive a large image, crop it into tiles, leaving some overlap between them, and then call the Python Mesmer code through the Reticulate R package (USHEY; ALLAIRE; TANG, 2022). We also implemented filtering to remove the smaller objects, which correspond to the image noise. Additionally, in every segmentation, we removed the objects that touched the border of the tile because they were probably divided when cropping the tiles and might not represent the actual cell. As there is an overlap between tiles, this removal does not prevent the nuclei from being detected.

5.3 MORPHOLOGICAL INFORMATION

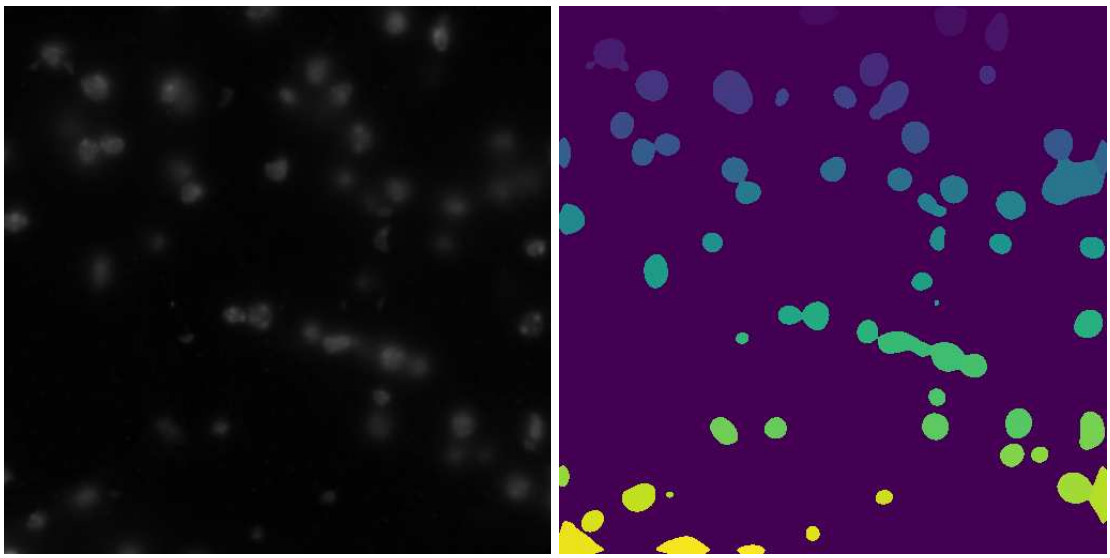
The next step after segmenting the images is to extract their morphological information. First, we tested the VAMPIRE algorithm. This Python package is open

source and available on GitHub. It was designed for Windows, but can be installed in Mac (UNIX-based operating system) through the usage of pip. We could not install it on Linux due to a dependency conflict in the library. The vampireanalysis package requires NumPy to be above a specific version. However, it depends on a package that requires NumPy to be below that version.

We also tried to install the wnd-charm package. It has two implementations: one command-line program written in C++ and a Python source code. Since we aim to integrate the tool with the laboratory R package, we opted to test the Python version. We followed the instructions on the website to build the code, but we were not able to install it. Since the package was released in 2008, it has not been properly maintained and we could not fix the installation error. We contacted our colleagues from the department, who had installed the software, and they recommended using Python 2.7. However, even after creating a virtual environment and following his instructions, we were unable to make the code work.

Lastly, we used the analyze particles tools present in the Fiji interface to get the features from the masks. To test the image, we selected five tiles from the DAPI image. Figure 47 presents one of these tiles and its mask. In this case, we did not remove the smaller objects and neither the nuclei around the borders to see how they would impact the data.

Figure 47 – One of the tiles and its mask used to extract the morphological information.

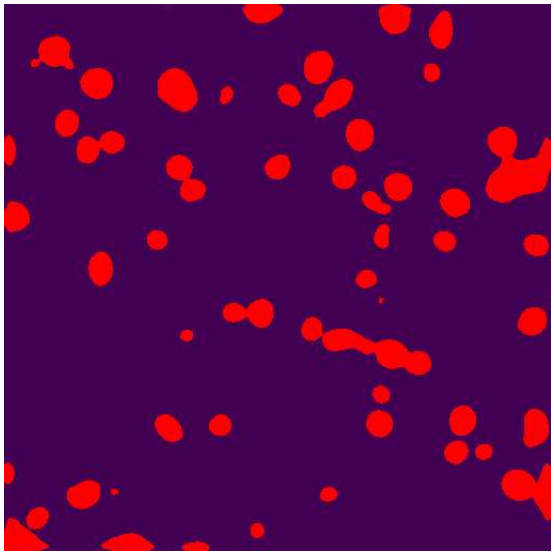


Source: Personal archive.

Although the Mesmer algorithm already presents the segmentation mask, the Fiji software is not able to recognize them. It requires the mask to be in the binary format, not in colors. So, we converted the mask to binary by applying the Huang color thresholding method. Figure 48 presents the mask obtained from the binarization. It is

possible to see that the algorithm merged some of the nuclei, thus there are some blobs with two or three cells, which can mislead the classification by creating objects with a greater area. Yet, no better solution was found in the Fiji platform and these groups were kept in the analysis.

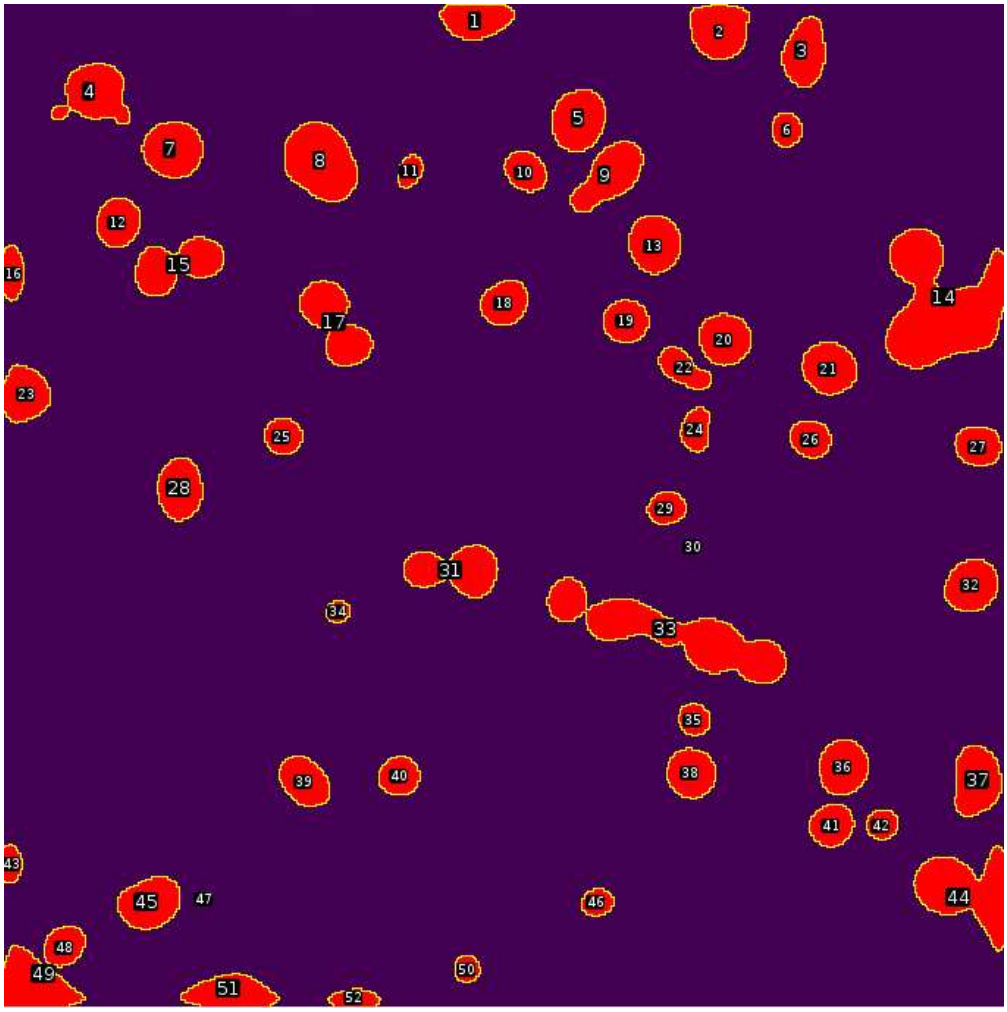
Figure 48 – Binarization obtained from Fiji's Huang Color Threshold method.



Source: Personal archive.

We extracted the morphological information by applying Fiji's analyze particles method. This tool returns both the image identifying each particle (Figure 49) and a table with the data for each cell. In this case, it is possible to see that objects 14, 33, 44, and others contain more than one cell and may skew the data.

Figure 49 – Particles analyzed by Fiji.

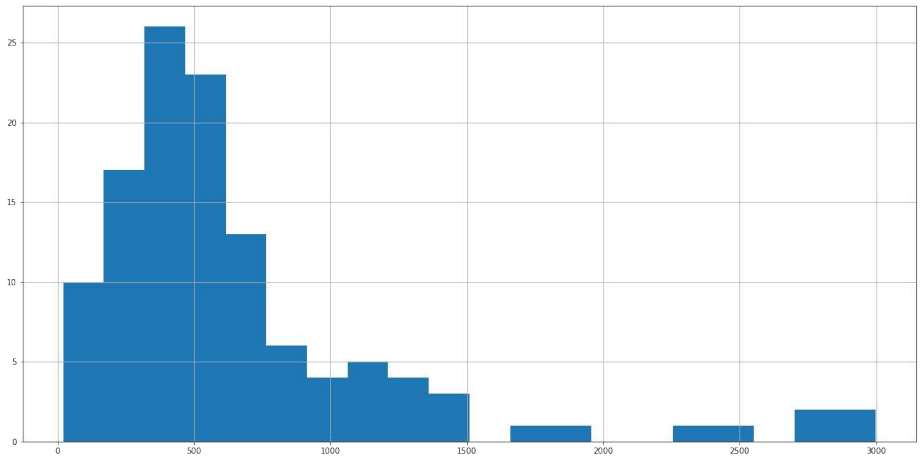


Source: Personal archive.

From the features present in Fiji, we opted to choose those that relied on the shape of the cell and could be easily interpreted. The software provided one matrix with the value for each feature for the particles in the image. Each image had around 60 particles on it, we merged the matrices for all tiles, obtaining 309 observations. Then, we removed the values that relied on the particle coordinates (X and Y, their mean, and base values) and all the rows that contained NAs, resulting in 119 particles.

We analyzed the histogram of the areas in the table (Figure 50) and noticed that there are some values in the right end. Those are since some cells were merged during the binarization and this must be filtered to avoid misleading results.

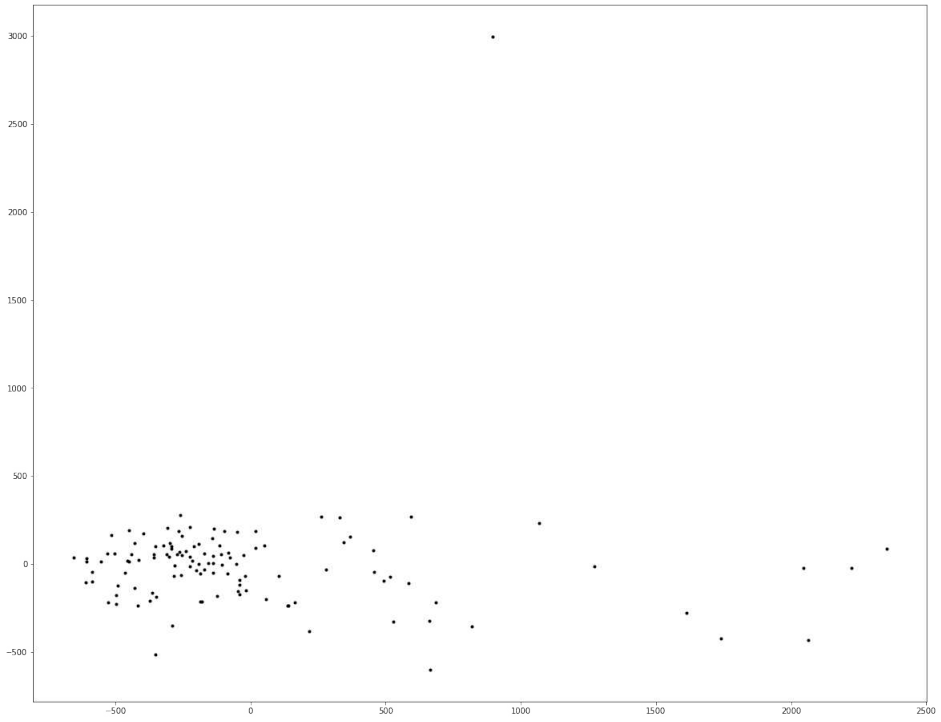
Figure 50 – Histogram of the area of the particles.



Source: Personal archive.

To see if the features were meaningful to the classification of the cell types, we used the PCA dimensionality reduction technique to visualize the data in two dimensions. Figure 51 shows how the data is dispersed. There are some outliers in the graph, however, it is not possible to identify clear clusters.

Figure 51 – Data in PCA reduced dimension.

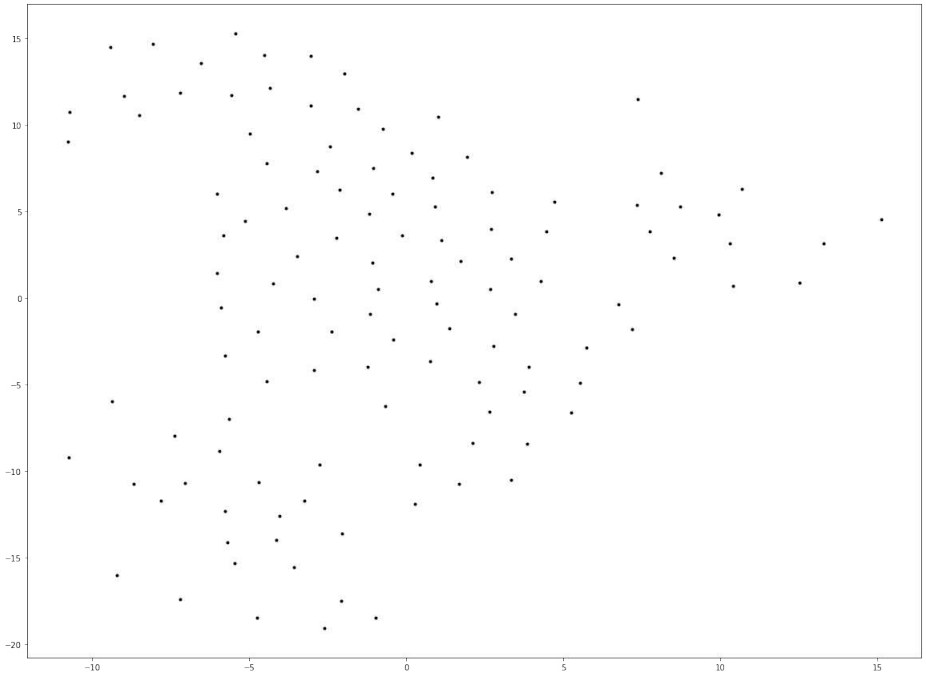


Source: Personal archive.

To further understand this issue, we also analyzed the data using the tSNE

technique (Figure 52). The data is better organized compared to the PCA visualization, but it is not clear to identify the clusters, which may lead to creating artificial ones. Further investigation is required to obtain a conclusion about the features obtained from the Fiji software.

Figure 52 – Data in tSNE reduced dimension.



Source: Personal archive.

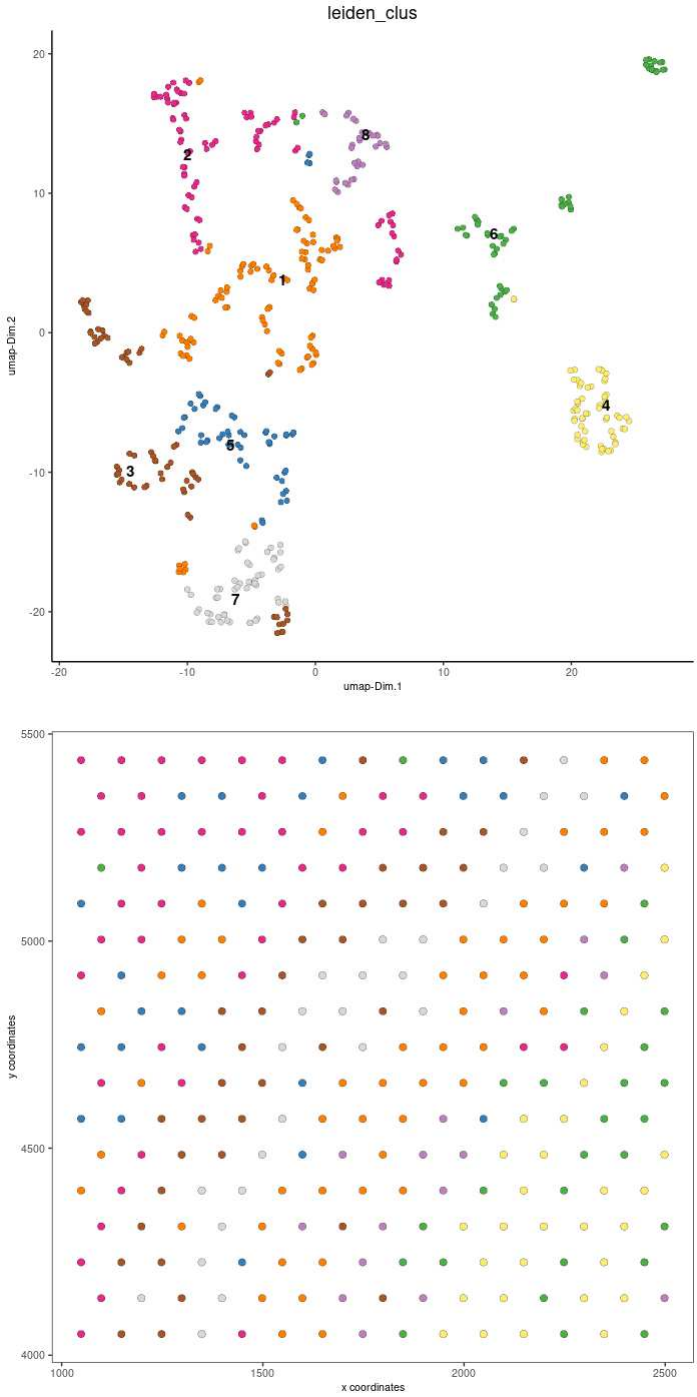
After realizing that the morphological information obtained from the Fiji tool was not conclusive, we looked for alternatives to obtain it. Instead of extracting the features directly from the masks, we considered converting them to polygons using the Giotto software to extract their characteristics. We tested the pyshp Python package to read the polygons files created in the R environment, which worked as expected. Then, we tried to implement the algorithm to measure the similarity between polygons and it did not work. First, there were some errors in the source files, which were fixed. Then, we were able to run the source code but got errors when calling the functions. There are other options to extract features from the polygons, but due to scheduling constraints, we could not implement them.

5.4 CLASSIFICATION

The idea behind the classification was to train a supervised algorithm that received as training data the cells that were in spots of pure cell types to then predict for those of mixed types. Therefore, the first step was to obtain the cell types and perform

their deconvolution. Using the Giotto software, we loaded the expression data from the spots, processed, and clustered them using Louvain's method. Figure 53 presents the clusters in the UMAP (top) and spatial dimension (bottom).

Figure 53 – Leiden clustering of the spatial data.

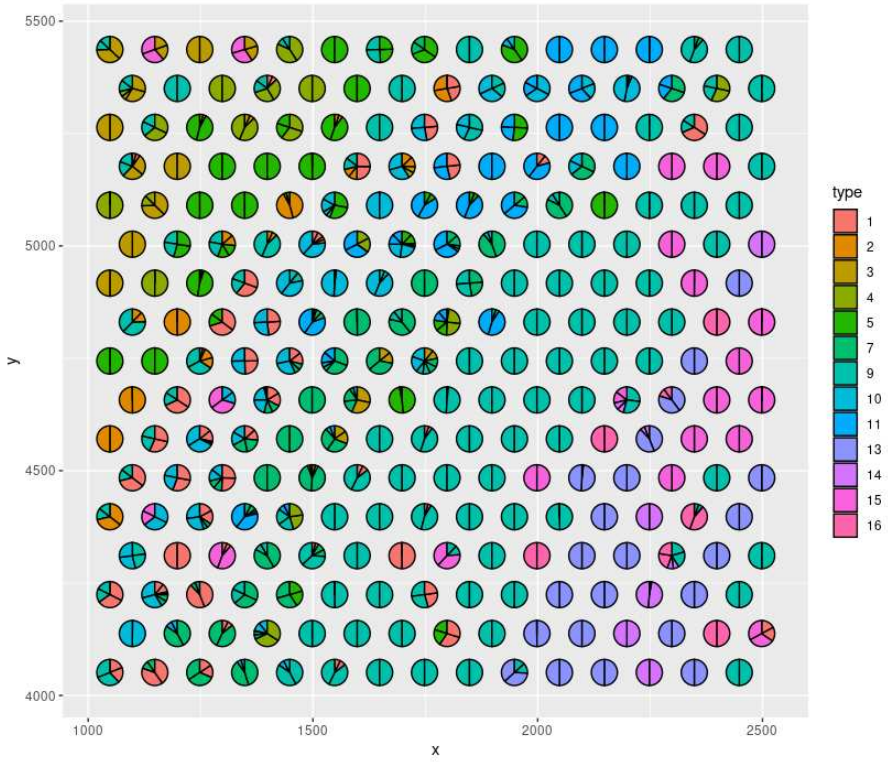


Source: Personal archive.

With this information, we applied cell-type deconvolution. The SpatialDWLS data requires the single-cell gene expression matrix of the same region to perform the

deconvolution. In this case, the MERFISH already presented this data. So we filtered this matrix to get the expression from the same region from where we created the pseudo-ST. Figure 32 presents the results from the cell-type deconvolution. Comparing this image with Figure 54, we can see that the number of clusters is different. This happens because the SpatalDWSL relies on the clusters that were identified in the single-cell data used to assist the algorithms. Therefore, as we used the Fusca package to calculate them, they do not match the ones obtained by the Giotto tool.

Figure 54 – Results from the cell-type deconvolution.



Source: Personal archive.

6 ANALYSIS

In this chapter, we analyze the results of the project. We investigate whether the work presented in chapter 5 meets the requirements of chapter 4. Instead of considering the project result as a whole, we identify the tasks individually to see the extent to which it was accomplished.

The first functional requirement (FR) was to create an ST dataset using the MERFISH data. This task was accomplished as it met the two non-functional requirements (NFR): the expression matrix counted the expression of the transcripts in the spots and not of the cells, and it maintained the spatial characteristics of the 10X Visium technology. The only issue regarding this requirement is the time it takes to create the dataset. It took a few hours to create the data presented in this report, which was just a portion of the MERFISH transcripts. Yet, we did not focus on this part as we would only perform it once.

The second FR was segmentation. The quality of the results depended on the model we chose. In this case, we are only considering the Mesmer algorithm as it was the most consistent. By using the Terra package and a sliding window we met the first NFR which was to not overflow the memory. Our tool also met the second NFR because the Mesmer model performed well in all the tiles we tested, regardless of their cell density (the metrics for the model are available in the Mesmer paper (cite Mesmer)). Third, we were able to remove the noise by defining a minimum area for the nuclei. Since we could not contact a pathologist in time for this report, we left this parameter to be defined by the user. Regarding the time, the model is relatively fast even when run on a computer without GPU. Finally, the Mesmer algorithm proved to generalize well as presented in its article.

The third FR was the morphological information, which had three non-functional ones. This step was partially accomplished. We were not able to extract a large number of features (hundreds or thousands) because the algorithm that performed such tasks failed to install. Instead, we used the Fiji software that provided only sixteen features. Regarding the time NFR, the software did not take long to obtain the values. Last, most of the characteristics found by Fiji were interpretable, satisfying the third non-functional requirement. However, this part of the project presented three major issues:

- To use the analyze particles function, we needed to convert the colored mask from Mesmer to binary masks. In this process, some of the cells were merged and they needed to be removed to not skew the results. To solve this issue, we could create individual tiles for the cells that would be merged before binarizing them. However, this would significantly increase the time to process the image and not satisfy the third NFR.

- The Fiji software relies on a guided user interface (GUI) which is written in java and cannot be easily incorporated into our R or Python code, restraining the automation of the code. One possible solution would be to use Jython to call the commands from the GUI. Yet, this tool required the Fiji software to be installed in the user computer, and, differently from a dependency, we cannot require a specific version to be installed for the code to run, leaving the package susceptible to errors.

- Additionally, when analyzing the features individually and in the reduced dimension space, we did not note a significant difference between them, indicating that they may not be sufficient for the classification. There needs to be a deeper investigation on this topic, but the algorithm will likely need other features. One possible cause is that the DAPI file only provides the nuclei, which is not as informative as the cell shape.

Since we could not accomplish the third requirement, the fourth and fifth FRs were not developed. Although we generated the cell-type deconvolution using the Giotto and CellRouter package, the number of clusters obtained from each algorithm was considerably different, which required further investigation. Additionally, the software used to extract the morphological information could not be incorporated into the code for automation and did now allow the processing of a large number of tiles. Moreover, after analyzing the values of the features, we did not notice promising results and decided to focus on other areas of the project.

Finally, even though we did not meet all functionalities of the project, we could test the generalization of the ones we developed. Given how well the Mesmer algorithm performed, we used it in the HE image by selecting the color channel that represented the nuclei and the results were equally satisfactory. Additionally, given the masks from the model, the process to extract the morphological information was analogous to the pseudo-ST. The other steps were not concluded for the pseudo-ST data, therefore they were not tested for the original ST dataset and were left for future work.

7 CONCLUSION

The field of spatial omics allows the comprehension of tissue architecture at an unforeseen level. However, this is the latest of the sequencing technologies and, while others had time to mature and evolve, SRT had just started. Therefore, these techniques lack perfection and have limitations. One of these is the balance between resolution and high throughput. The 10X Visium technology, one of the major technologies available today, can collect expression information from the whole transcriptome but is not able to get this data from single cells. Having a clear understanding of genomics and location could lead to discoveries in different fields of biology.

In this project, we proposed an algorithm to enhance the resolution of ST by assigning the cell type to its location. The method was designed to have three major parts: segmentation, extraction of morphological information, and classification. Additionally, these items would be tested in a pseudo-ST created from the MERFISH dataset. Due to software limitations and time constraints, the second task did not produce satisfactory results, becoming a hindrance for the third one. Nevertheless, the work was fruitful as the segmentation algorithm created here could be extended to other projects in the group. Additionally, the limitations found here can indicate deficiencies in current algorithms, leaving room for code development in the future.

Regarding the actual development, the first task tested different algorithms for segmentation and noted that most of them failed in denser regions of the image, the exception was the Bernsen local thresholding and Mesmer method. We chose the last one to be the core in our segmentation function due to its generality and robustness to noise. Our final tool encompassed a sliding window function that was able to receive a large tissue image, apply a sliding window, segment the nuclei in each tile and also remove the noise. This method works with the DAPI image from the pseudo-ST, and also with the HE data from the ST. This tool proved to be useful in other projects of the laboratory. Therefore, we incorporated it into our package for spatial omics analysis, contributing to the Giotto software and the field as a whole. It is now open-source and available on GitHub.

The second task relied on extracting morphological information from the masks obtained in the segmentation. We verified different packages and they presented problems either on the installation or when calling the function. Indicating that there is a lack of functional packages available in this area, and also leaving space for future projects. Therefore, instead of using a Python or R library, we used the Fiji GUI. This tool provided interpretable features, but it was limited regarding reading the data, automating the process, and providing relevant features. Further investigation is required on how to overcome these issues, but due to time restrictions. The unsatisfactory results obtained in this item hindered the development of the classification tool.

In conclusion, the work presented here provided a tool for nuclei segmentation in both DAPI and HE images. Moreover, it shed light on the limitations of current techniques to extract morphological features from the cells. Lastly, it gave direction on what the next steps of the project should be. We aim to continue working on enhancing the resolution of ST, taking a closer look at the tools to extract morphological information and how to improve them.

BIBLIOGRAPHY

ASP, Michaela; BERGENSTRÅHLE, Joseph; LUNDEBERG, Joakim. Spatially Resolved Transcriptomes—Next Generation Tools for Tissue Exploration. **BioEssays**, Wiley, v. 42, n. 10, p. 1900221, May 2020. DOI: 10.1002/bies.201900221. Available from: https://doi.org/10.1002/bies.201900221.

BURGESS, Darren J. Spatial transcriptomics coming of age. **Nature Reviews Genetics**, v. 20, n. 6, p. 317–317, June 2019. ISSN 1471-0064. DOI: 10.1038/s41576-019-0129-z. Available from: https://doi.org/10.1038/s41576-019-0129-z.

CHAN, John K. C. The Wonderful Colors of the Hematoxylin–Eosin Stain in Diagnostic Surgical Pathology. **International Journal of Surgical Pathology**, v. 22, n. 1, p. 12–32, 2014. PMID: 24406626. DOI: 10.1177/1066896913517939. eprint: https://doi.org/10.1177/1066896913517939. Available from: https://doi.org/10.1177/1066896913517939.

DONG, Rui; YUAN, Guo-Cheng. SpatialDWLS: accurate deconvolution of spatial transcriptomic data. **Genome Biology**, v. 22, n. 1, p. 145, May 2021. ISSN 1474-760X. DOI: 10.1186/s13059-021-02362-7. Available from: https://doi.org/10.1186/s13059-021-02362-7.

DRIES, Ruben et al. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. **Genome Biology**, v. 22, n. 1, p. 78, Mar. 2021. ISSN 1474-760X. DOI: 10.1186/s13059-021-02286-2. Available from: https://doi.org/10.1186/s13059-021-02286-2.

FAN, Hongchao; ZHAO, Zhiyao; LI, Wenwen. Towards Measuring Shape Similarity of Polygons Based on Multiscale Features and Grid Context Descriptors. **ISPRS International Journal of Geo-Information**, v. 10, n. 5, 2021. ISSN 2220-9964. DOI: 10.3390/ijgi10050279. Available from: https://www.mdpi.com/2220-9964/10/5/279.

GREENWALD, Noah F. et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. **Nature Biotechnology**, Nov. 2021. ISSN 1546-1696. DOI: 10.1038/s41587-021-01094-0. Available from: https://doi.org/10.1038/s41587-021-01094-0.

HIJMANS, Robert J. **terra: Spatial Data Analysis**. [S.l.], 2022. R package version 1.5-21. Available from: https://CRAN.R-project.org/package=terra.

HU, Jian; SCHROEDER, Amelia; COLEMAN, Kyle; CHEN, Chixiang; AUERBACH, Benjamin J.; LI, Mingyao. Statistical and machine learning methods for spatially resolved transcriptomics with histology. **Computational and Structural Biotechnology Journal**, v. 19, p. 3829–3841, 2021. ISSN 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2021.06.052. Available from: https://www.sciencedirect.com/science/article/pii/S2001037021002907.

KAPUSCINSKI, Jan. DAPI: a DNA-Specific Fluorescent Probe. **Biotechnic & Histochemistry**, Taylor & Francis, v. 70, n. 5, p. 220–233, 1995. PMID: 8580206. DOI: 10.3109/10520299509108199. eprint: https://doi.org/10.3109/10520299509108199. Available from: https://doi.org/10.3109/10520299509108199.

LANGER-SAFER, P. R.; LEVINE, M.; WARD, D. C. Immunological method for mapping genes on Drosophila polytene chromosomes. eng. **Proceedings of the National Academy of Sciences of the United States of America**, v. 79, n. 14, p. 4381–4385, July 1982. PMC346675[pmcid]. ISSN 0027-8424. DOI: 10.1073/pnas.79.14.4381. Available from: https://doi.org/10.1073/pnas.79.14.4381.

MARX, Vivien. Method of the Year: spatially resolved transcriptomics. **Nature Methods**, Springer Science and Business Media LLC, v. 18, n. 1, p. 9–14, Jan. 2021. DOI: 10.1038/s41592-020-01033-y. Available from: https://doi.org/10.1038/s41592-020-01033-y.

MOFFITT, J.R.; ZHUANG, X. Chapter One - RNA Imaging with Multiplexed Error-Robust Fluorescence In Situ Hybridization (MERFISH). In: FILONOV, Grigory S.; JAFFREY, Samie R. (Eds.). **Visualizing RNA Dynamics in the Cell**. [S.l.]: Academic Press, 2016. v. 572. (Methods in Enzymology). P. 1–49. DOI: https://doi.org/10.1016/bs.mie.2016.03.020. Available from: https://www.sciencedirect.com/science/article/pii/S0076687916001324.

ORLOV, Nikita; SHAMIR, Lior; MACURA, Tomasz; JOHNSTON, Josiah; ECKLEY, D. Mark; GOLDBERG, Ilya G. WND-CHARM: Multi-purpose image classification using compound image transforms. eng. **Pattern recognition letters**, v. 29, n. 11, p. 1684–1693, Jan. 2008. PMC2573471[pmcid]. ISSN 0167-8655. DOI: 10.1016/j.patrec.2008.04.013. Available from: https://doi.org/10.1016/j.patrec.2008.04.013.

PHILLIP, Jude M.; HAN, Kyu-Sang; CHEN, Wei-Chiang; WIRTZ, Denis; WU, Pei-Hsun. A robust unsupervised machine-learning method to quantify the morphological heterogeneity of cells and nuclei. **Nature Protocols**, v. 16, n. 2, p. 754–774, Feb. 2021. ISSN 1750-2799. DOI: 10.1038/s41596-020-00432-x. Available from: https://doi.org/10.1038/s41596-020-00432-x.

ROCHA, Edroaldo Lummertz da et al. Reconstruction of complex single-cell trajectories using CellRouter. **Nature Communications**, Springer Science and Business Media LLC, v. 9, n. 1, Mar. 2018. DOI: 10.1038/s41467-018-03214-y. Available from: https://doi.org/10.1038/s41467-018-03214-y.

ROSAI, Juan. Why microscopy will remain a cornerstone of surgical pathology. **Laboratory Investigation**, v. 87, n. 5, p. 403–408, May 2007. ISSN 1530-0307. DOI: 10.1038/labinvest.3700551. Available from: https://doi.org/10.1038/labinvest.3700551.

SCHINDELIN, Johannes et al. Fiji: an open-source platform for biological-image analysis. **Nature Methods**, v. 9, n. 7, p. 676–682, July 2012. ISSN 1548-7105. DOI: 10.1038/nmeth.2019. Available from: https://doi.org/10.1038/nmeth.2019.

SCHMIDT, Uwe; WEIGERT, Martin; BROADDUS, Coleman; MYERS, Gene. Cell Detection with Star-Convex Polygons. **Lecture Notes in Computer Science**, Springer International Publishing, p. 265–273, 2018. ISSN 1611-3349. DOI: 10.1007/978-3-030-00934-2_30. Available from: http://dx.doi.org/10.1007/978-3-030-00934-2_30.

STÅHL, Patrik L. et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. **Science**, American Association for the Advancement of Science (AAAS), v. 353, n. 6294, p. 78–82, June 2016. DOI: 10.1126/science.aaf2403. Available from: https://doi.org/10.1126/science.aaf2403.

STRINGER, Carsen; WANG, Tim; MICHAELOS, Michalis; PACHITARIU, Marius. Cellpose: a generalist algorithm for cellular segmentation. **Nature Methods**, v. 18, n. 1, p. 100–106, Jan. 2021. ISSN 1548-7105. DOI: 10.1038/s41592-020-01018-x. Available from: https://doi.org/10.1038/s41592-020-01018-x.

USHEY, Kevin; ALLAIRE, JJ; TANG, Yuan. **reticulate: Interface to 'Python'**. [S.l.], 2022. R package version 1.24. Available from: `https://CRAN.R-project.org/package=reticulate`.