Image Analysis of Circulating Tumour Cell Clusters from Imaging Flow Cytometry Data

Master's thesis in Biomedical Engineering

Filip Berg

December 2020

LUND UNIVERSITY

LTH
FACULTY OF ENGINEERING

Department of Biomedical Engineering

Main supervisor: Per Augustsson
Co-supervisors: Cecilia Magnusson & Pontus Nordenfelt
Examiner: Thomas Laurell

To my father

Abstract

Circulating tumour cells (CTCs) are cancer cells that have entered the circulation of the body breaking free from their primary tumour and that can act as progenitors of metastasis. At the time of writing, a study on a novel method to detect and count CTCs using imaging flow cytometry (IFC) is being conducted at Lund University. In the study, a problem was found where CTCs clustered with normal white blood cells (WBCs) were not detected as CTC candidates. These CTCs were not detected because the analysis software treated clusters the same as single cells. The rarity of CTCs in blood means it is important to detect every single one in a sample.

This thesis aimed to develop an algorithm that could detect CTC - WBC clusters in IFC data of prostate cancer patient samples. An algorithm that could automate the detection of CTC candidates would simplify the present process which suffer from excessive manual assessment. The main problem to be solved was to segment the different cells in the clusters from each other in the images.

An algorithm to detect CTC - WBC clusters in IFC data was proposed and was initially tested on three patient datasets. The algorithm showed stable segmentation results. The problem of segmenting cells was solved by using an Otsu threshold and watershed approach on images of cells stained with the nucleic fluorescent marker DAPI. The segmented regions could then be used to examine the fluorescent intensity of other stains within the regions. The initial results of CTC detection were promising. The number of candidates to manually assess to find CTC - WBC clusters was greatly reduced and is now at a manageable level.

At the time of writing this, the program is deployed and ready for use in the continuation of the study.

Keywords: MSc, Image Analysis, Segmentation, Otsu Threshold, Watershed, Circulating Tumour Cells, Imaging Flow Cytometry

Acknowledgements

Firstly, I would like to thank my principal supervisor Per Augustsson for successfully guiding me through this thesis. I would also like to thank my co-supervisors; Pontus Nordenfelt for discussions on image analysis, and Cecilia Magnusson for discussions on biology as well as assessing my results.

Contents

1 Introduction **7**

1.1 Background Study 8

1.1.1 CellSearch CTC Test 9

1.1.2 New IFC Method 9

1.2 Definition of CTC 10

1.3 Problem Statement 10

1.4 Aim 10

2 Theory **11**

2.1 Imaging Flow Cytometry 11

2.2 Fluorescent Dyes 14

2.2.1 DAPI 14

2.2.2 panCK 14

2.2.3 EpCAM 14

2.2.4 CD45/CD66b 15

2.2.5 Auto-Fluorescence 15

2.3 Image Analysis 16

2.3.1 Segmentation and Thresholding 16

2.3.2 Otsu Thresholding 17

2.3.3 Binary Morphological Operations 19

2.3.4 Watershed 21

2.3.5 Gaussian Filter 25

3 Methods **27**
3.1 Data Acquisition 27
3.2 Datasets 29
3.3 Algorithm Development 30

4 Results **33**
4.1 Overview of Algorithm 33
4.1.1 Determining Thresholds 36
4.2 Examples of Detected Events 36
4.3 Segmentation 39
4.4 Detection of CTCs from Datasets 40
4.4.1 Patient 6 41
4.4.2 Patient 8 42
4.4.3 Patient 9 43
4.5 Graphical User Interface 44

5 Discussion **47**
5.1 Segmentation 47
5.2 CTC Detection 49
5.2.1 False Positives 50
5.2.2 Determining Thresholds 50
5.3 Recommended Use of Algorithm 51
5.4 Future work 52

6 Conclusion **53**

References **55**

Chapter 1

Introduction

Circulating tumour cells, or CTCs, are cancer cells that have entered the circulation of the body breaking free from their primary tumour. These cells have the potential to migrate into other tissue and act as progenitors of metastasis. [1] Survivability generally decreases substantially when cancer has the ability to metastasize. [2]

Since CTCs are a precursor to metastasis, the number of CTCs in blood could act as a marker for cancer progression. In fact, a study published in 2004 concluded that the number of CTCs in patients before treatment was a predictor in survivability of metastatic breast cancer. [3]. Thus, a reliable method to measure the number of CTCs is of great interest, both diagnostically and therapeutically. This thesis attempts to reduce manual labour and partly automate a step in a novel method to identify and count CTCs in blood samples.

Currently, there is only one approved method to count CTCs used clinically, the CellSearch® CTC Test, which is commercially available. This test was FDA approved in January 2004. The first stage of this method is to separate cells that are positive to an epithelial cancer cell marker called EpCAM using immunomagnetic beads. However, studies have shown that far from every epithelial CTC express EpCAM. [4, 5] Further, CTCs of non-epithelial origin cannot be detected.

As CTCs are extremely rare, they are difficult to detect in patient blood samples. Failing to detect just a few CTCs could have an impact on test results.

Typically, CTCs attribute to less than 1 cell per 10^5 to 10^6 mononuclear cells in blood. [6] This ratio would pose a great challenge on any detection method.

Advancements in technology have led to new opportunities in developing better methods. The proposed novel method is using microfluidic CTC enrichment chips and imaging flow cytometry, or IFC. IFC is a powerful method providing image data on fluorescent markers as well as morphology of cells one by one. [7] This method should have the advantage of being more general because it does not initially separate cells based on one marker.

At present, the IFC data is analysed in proprietary software where a problem has been identified. CTCs clustered with white blood cells (WBCs) are sorted out and it is not feasible to manually find them. Thus, an automatic method of detecting these CTC - WBC clusters is necessary.

It is known from literature that CTCs form clusters with themselves and that they can also form clusters with WBCs. [8] A 2019 study in Nature suggests that CTC-neutrophil clustering, the most common type of CTC-WBC-interaction, increases the metastatic potential of the CTCs. [9] This makes CTC clustering interesting to detect and study further.

In this thesis, a specialized algorithm pipeline was developed in MATLAB® to detect CTCs in imaging flow cytometry data of prostate cancer patient samples. Specially, to aid the detection of clusters of CTCs and WBCs. As every dataset is different with regards to the intensity of fluorescent markers, the main goal was to limit the number of cell images needed to manually assess. A nuclear image segmentation approach is proposed with the intensities of the fluorescent markers then analysed within these regions. A graphical user interface was also created to ease the use of the algorithm.

1.1 Background Study

The background to this thesis was a study conducted by Cecilia Magnusson at the department of biomedical engineering and the department of translational medicine collaboratively at Lund University. At the time of writing this the study is still on-going. The purpose of the study is to develop and assess the viability of an alternative method to detect CTCs using microfluidic CTC-chips and imaging flow cytometers. The way the method is initially evaluated is by direct comparison to that of the only approved method, the CellSearch CTC test.

The long-term goal of the study aims to develop a general purpose CTC detection method that can detect CTCs based on any cancer markers in conjunction. The main benefit being not including an initial enrichment step based on

a sole marker.

1.1.1 CellSearch CTC Test

The CellSearch method enriches CTCs from blood by separating EpCAM positive cells using immunomagnetic beads. These are anti-EpCAM antibodies attached to ferrofluid nanoparticles that can be magnetically manipulated. EpCAM is an epithelial cancer marker. The separated cells are then fluorescently stained with cytokeratin antibodies, DAPI and CD45. Cytokeratin is an epithelial cancer cell marker, DAPI is a cell nucleus marker and CD45 is a WBC marker. The cells are then imaged in fluorescence microscopy and are manually assessed. A CTC is defined as cytokeratin positive, DAPI positive and CD45 negative (as well as EpCAM positive). The reason for CD45 inclusion is WBC contamination in the initial separation step. The CellSearch CTC test is a commercial method.

Problem The main problem with the CellSearch method is that it separates EpCAM positive cells in the initial step. All EpCAM negative cells are then discarded. Research has shown that epithelial tumour cells might undergo so-called epithelial-to-mesenchymal transition, or EMT. [4] This is a sort of stem cell-like transition. EMT causes an epithelial tumour cell to lose its epithelial characteristics, such as losing expression of EpCAM. This means such CTCs are not detected. Also, the CellSearch method cannot be used on non-epithelial CTCs.

1.1.2 New IFC Method

The method proposed in the study enriches CTCs by utilizing a microfluidic acoustophoresis CTC enrichment chip. [10] This CTC chip separates CTCs by their acoustic properties. All remaining cells can then be fluorescently stained using any markers of choice and then imaged in an imaging flow cytometer. If the same markers are used as in the CellSearch method then the main benefit is that it is possible to detect EpCAM or panCK positive cells separately. See section 3.1 for a more in depth summary of the method.

Problem The price paid using the CTC chip is that smaller CTCs are sorted out along with WBCs. This makes it necessary to allow for some WBCs to contaminate the sample. The numbers of WBCs are orders of magnitude larger than CTCs which means that they will produce a lot of irrelevant data in IFC. This calls for robust image analysis methods to discriminate CTCs from WBCs.

1.2 Definition of CTC

In this thesis a CTC is defined as a cell that conforms to the following criteria:

1. DAPI positive (has nuclei),
2. CD45/CD66b negative (is not a WBC) and
3. EpCAM **or** panCK positive (has epithelial cancer marker).

Please note the fact that CTCs are defined as either EpCAM or panCK positive. If a cell expresses both EpCAM and panCK it is here called classic CTC, referring to the fact that it would be detected in the CellSearch CTC test.

In addition, CTCs have single lobe nuclei while some WBCs have fragmented nuclei or non-circular shapes. This last step is done qualitatively both in the CellSearch method and in the new proposed method. It is not specifically addressed in this thesis.

1.3 Problem Statement

The problem addressed in this thesis is that CTCs and WBCs form clusters and cannot be detected. This is because the current software does not differentiate between fluorescence coming from different cells in the IFC images. This essentially means that it treats clusters as single cells. When a CTC and a WBC lies side by side in an image the WBC produces positive signal in the CD45/CD66b channel. This causes the software to sort out the cell based on criteria 2 in section 1.2. This is of course not acceptable. The main problem to solve in this thesis was to segment the cells in the images from each other. Then, the fluorescent intensities of each single cell can be examined on its own.

1.4 Aim

The aim of this thesis was to construct a program that could reduce the number of images of cells needed to be manually assessed to find the clusters mentioned in section 1.3 by detecting CTC candidates. A finished program were then to be used in the continuation of the study by Cecilia Magnusson.

Chapter 2

Theory

2.1 Imaging Flow Cytometry

In the last 10-15 years imaging flow cytometry has emerged as a standard piece of equipment in medical laboratories. Imaging flow cytometry can be described as a marriage between conventional flow cytometry and fluorescence imaging. In contrast to conventional cytometry it supplies images of the objects. This allows for far more complex analysis of morphology and signal origin. Of which, in theory, could be automated by image analysis methods. On the other hand, this results in larger comprehensive datasets putting its demands on the analysis. Imaging flow cytometers can capture brightfield and multiple fluorescence images of thousands of cells in the matter of hours, which means it enables studies of rare cell populations, e.g. CTCs. [11]

Flow cytometry and fluorescence imaging both spring from the same underlying concept, which is to label cells by conjugates of antibodies and fluorophores. Both approaches combines the specificity of antibodies to specific molecular targets and the detectability of fluorophores. By using specific molecular properties, different cell types can be differentiated. [12] The difference is that fluorescence microscopy is typically a qualitative technique using relatively few cells. Flow cytometry, on the other hand, is unquestionably a quantitative technique that can count thousands of cells per second and reliably provide

repeatable results. [12] It is, of course, possible to construct fluorescence microscopy systems with high throughput. However, flow cytometry and imaging flow cytometry are well suited for cells in suspension, e.g. blood samples.

Fluorescence When a fluorophore absorbs a photon the molecule's energy state moves from the ground state to an excited state. This excess energy is dissipated by emitting another photon. This other photon has slightly lower energy than the original photon due to energy being lost in the process. [13] Because the light leaving the excited molecule is slightly red-shifted, one can differentiate between the excitation light and the fluorescent light.

Fluorescence Microscopy In fluorescence microscopy a sample is illuminated with a specific wavelength of light, which gives rise to fluorescence of a slightly longer wavelength. This much weaker light is then separated by filters in the optics. Ideally, only fluorescent light should pass through the filters to the observer shown on a dark background. [13] The downside of this technique is that the analysis is often time consuming and is prone to user bias. [12]

Flow Cytometry In flow cytometry, cells are focused to the center of a microfluidic channel, in a process called hydrodynamic focusing, ideally passing one by one through the sensing zone. In the sensing zone, the cells are exposed to focused laser light, which give rise to fluorescence and scattered light. Optics can then pick up these signals at different wavelengths and finally measure cell properties. [12] The downside to this technique is that it provides no morphological or spatial information. Flow cytometry is today used diagnostically for many diseases. [11]

Imaging Flow Cytometry Imaging flow cytometry combines the high throughput and high level of automation offered by flow cytometry with the qualitative aspects of fluorescence imaging. This makes it a powerful technique able to capture data inaccessible with each individual technique.

As of 2017 there were only two imaging flow cytometry systems available, the Amnis Flowsight and the Amnis ImageStream. As the cells pass the sensing area they can be imaged in twelve different ways, two brightfield images and up to ten fluorescent or darkfield images. [12] Brightfield measures the transmittance of light through a sample. The background is thus bright and the specimen appears dark. Darkfield is the opposite, measuring light at an angle from the light direction detection only scattered light. The background is thus dark and areas where light scatters of the specimen appear bright.

The images share the same problem with time consumption and user bias in analysis. However, there is a potential to much more easily automate the process using image analysis techniques. This is because it is much easier to locate and segment single cells in a channel rather than hundreds of randomly positioned cells on a microscopy slide. [14] Of course, problems can arise when two or more cells have clustered, which is the issue addressed in this thesis.

As of today imaging flow cytometry is usually not used to its full potential. The produced images contain high degrees of morphological and structural data but are usually analysed by only a few hand-selected features and often by applying subjective binary gates that reduces the information content enormously. [11] The Amnis devices today come with the proprietary analysis software IDEAS. This software provides automated segmentation and for example allows for setting gates for selected features.

2.2 Fluorescent Dyes

Fluorescent staining is the principal component of fluorescent imaging. It uses the specificity of specially designed molecules, e.g. antibodies (immunofluorescence), to bind to specific target molecules. This means they will co-localize with the target to be detected. The molecule is either itself fluorescent or has a fluorophore attached to it. The core function of the fluorophore is, after absorbing laser light produced by the instrument, to emit red-shifted light of a specific wavelength. [15] This light can then be detected by a camera sensor in an optical microscope or by a photodetector in the case of a flow cytometer. The fluorescent dyes used in this work are presented below. As most cancers are of epithelial origin [16], the cancer markers used here are essentially epithelial markers. This way, cancer cells can be differentiated from WBCs and other blood cells that do not express these markers.

2.2.1 DAPI

DAPI, 4,6-diamidino2-phenylindole, is a molecule that binds to DNA forming a fluorescent complex. [17] As most DNA exists within the nucleus, DAPI can be used as nuclear staining to visualize nuclei in fluorescent imaging.

2.2.2 panCK

panCK, or pan anti-cytokeratin, is a type of antibody designed to target human cytokeratin proteins. Cytokeratins are proteins forming parts of the cytoskeletal intermediate filaments in epithelial tissue. The panCK used in this study detects cytokeratin 4, 5, 6, 8, 10, 13 and 18. The exact expression of different cytokeratins vary with tissue type. These filaments are found throughout the cytoplasm and should be detected over the whole cell. When epithelial tissue turns cancerous, cytokeratin expression is usually unchanged and thus used as an epithelial cancer cell marker. [18] In the study, panCK was conjugated with the fluorophore AF-488.

2.2.3 EpCAM

EpCAM, or epithelial cell adhesion molecule, is a transmembrane glycoprotein. Antibodies targeting EpCAM are thus found on the surface of EpCAM-positive cells. EpCAM has many proposed functions, the main being involved in adhesion between adjacent epithelial cells. [19] Nearly all cancers derived from epithelial cells express EpCAM [20] and is thus an excellent epithelial cancer cell

marker. In the study the EpCAM antibody was conjugated with the fluorophore phycoerythrin (PE).

2.2.4 CD45/CD66b

CD45 is a transmembrane glycoprotein expressed by all WBCs and exclusively by cells of the hematopoietic system. [21] It has an important function in signalling immunological responses. [21] Antibodies targeting CD45 should therefore be found on the surface of WBCs and not be found on any CTC.

In the background study, CD45 was determined not adequate in staining all WBCs, in particular some granulocytes. This is because some granulocytes did not express enough CD45 to distinguish them from CTC auto-fluorescence. CTCs often exhibit high levels of auto-fluorescence and were in some cases indistinguishable from these WBCs. On that account, CD66b was added to later patient samples to bump up their signal. CD66b is a glycoprotein anchored to the extracellular membrane. [22] It has only been shown to be expressed in granulocytes. [23]

In the study, CD45 and CD66b antibodies were conjugated with the fluorophores allophycocyanin (APC) and AF-647 respectively, sharing the same excitation laser wavelength and emission spectra. This means the signal will be combined in immunofluorescent imaging.

2.2.5 Auto-Fluorescence

Cells can contain naturally occurring fluorophores, so-called endogenous fluorophores, and the fluorescence from these substances is called autofluorescence. [13] This can be a major nuisance and challenge in fluorescent imaging, introducing noise and false-positives in experiment data. In the study, the most notable cells exhibiting this phenomenon are eosinophils. Compared to other WBCs, they produce bright auto-fluorescence attributed to the molecule flavin adenine dinucleotide existing within eosinophil granules. [24] This causes eosinophils to fluoresce in every channel in the imaging flow cytometer, which provides a challenge when classifying the cells. It is also important to note that debris or non-cell material imaged by IFC have a tendency to auto-fluoresce as well.

2.3 Image Analysis

An image can be represented as discrete numbers in a matrix corresponding to intensity and location. This means one can apply mathematical operations to alter (filter) or extract information from images.

Another way to represent an image is with histograms. They show the distribution of pixel intensities in the image, although losing all spatial information. This is for example useful for thresholding that is described below.

2.3.1 Segmentation and Thresholding

Image segmentation is the operation of partitioning pixels in an image into regions. It is in fact a key problem in image analysis, often presenting bottlenecks for automated algorithms. The simplest of segmentations would be to determine foreground and background in an image, e.g. where there is signal and where there is not. When a region has been segmented one can extract useful information about it, such as shape, mean intensity and more. A more challenging example could be segmenting a car in a streetview image.

Image thresholding is one way to segment an image. In fluorescence imaging the region of interest is usually a light region on a dark background. In this case one wants to group pixels of similar intensity with each other. A simple way to do this is by thresholding. [25] Thresholding of an image, $i(x, y)$, yields the thresholded (binary) image, $b(x, y)$, with the threshold, T, conforming to the equation

$$b(x,y) = \begin{cases} 1, & \text{if } i(x,y) > T \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

One way to select T is by using Otsu's threshold. See figure 2.1 for an example of a thresholded image.

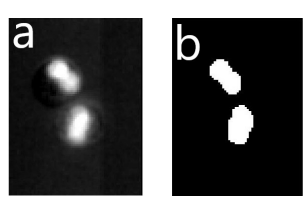

Figure 2.1: a) original image and b) thresholded image by Otsu's method. Image depicting nuclear DAPI staining of probably two clustered neutrophils from imaging flow cytometry data.

2.3.2 Otsu Thresholding

Otsu thresholding is an automatic threshold selecting method based solely on the histogram of a grayscale image. The method is commonly used global thresholding method in fluorescence microscopy. [26] It is a global thresholding method as it computes a single threshold based on all pixels in the image. The method was first proposed by Nobuyuki Otsu in 1979. [27] The output is a binary image of the same size as the original image.

In essence, the method maximizes the separability of two assumed classes in histograms (e.g. pixel intensity peaks in a bimodal histogram) by maximizing the between-class variance. [28] This means the method seeks to find a threshold such that the difference between the classes is the greatest. The method has the advantage of not requiring any prior knowledge of the data origin. Otsu thresholding can easily be implemented in MATLAB via in-built functions. See figure 2.1 for an example of an Otsu thresholded image.

Algorithm

The following simplified description of Otsu's algorithm is based on the original article. [27] At first, the normalized histogram, p_i, is computed using the pixel intensities $i = 1$ through L. L being the largest pixel value. The histogram then satisfies

$$1 = \sum_{i=1}^{L} p_i, \text{ where } p_i \geq 0. \tag{2.2}$$

Assuming a threshold T creates two classes, the background $C_0 = \{1, \ldots, T\}$ and foreground $C_1 = \{T + 1, \ldots, L\}$, please refer to figure 2.2. The probability of class C_0 is

$$P(C_0) = \sum_{i=1}^{T} p_i. \tag{2.3}$$

The probability of class C_1 is simply the statistical complement

$$P(C_1) = 1 - P(C_0). \tag{2.4}$$

The means of each class are computed via

$$\mu_0 = \sum_{i=1}^{T} \frac{i p_i}{P(C_0)} \tag{2.5}$$

and

$$\mu_1 = \sum_{i=T+1}^{L} \frac{i p_i}{P(C_1)}, \tag{2.6}$$

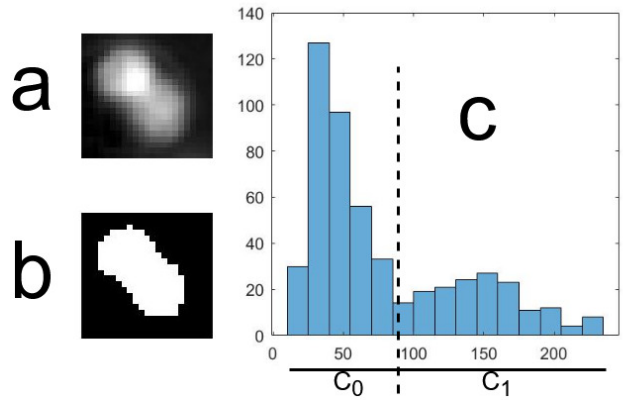

Figure 2.2: a) original image, b) otsu thresholded image and c) histogram of original image. An approximate otsu threshold, C_0 and C_1 have been marked.

which can be shown by Bayes' formula. The global mean ($P(C_0) = 1$) of the image is

$$\mu_G = \sum_{i=1}^{L} i p_i. \tag{2.7}$$

The between-class variance, well known in discriminant analysis, is defined as

$$\sigma_B^2 = P(C_0)(\mu_0 - \mu_G)^2 + P(C_1)(\mu_1 - \mu_G)^2. \tag{2.8}$$

The Otsu threshold, T^*, is obtained by computing σ_B^2 in eq. 2.8 for all possible integer thresholds T, $1 \leq T \leq L$, using the values computed in eq. 2.2-2.7. T^* is chosen to maximize between-class variance, or in mathematical terms

$$\sigma_B^2(T^*) = \max_{1 \leq T \leq L} \sigma_B^2(T). \tag{2.9}$$

It is worth noting that maximizing between-class variance is equivalent to minimizing within-class variance, σ_W^2, due to

$$\sigma_W^2 = \sigma_G^2 - \sigma_B^2, \tag{2.10}$$

where σ_G^2 is the global variance independent of chosen threshold T. Between-class variance is chosen due to less computational expense.

The algorithm presented here can be heavily optimised. It is also typically implemented as standard in image processing software.

2.3.3 Binary Morphological Operations

Morphological image processing is a collective name for techniques based on mathematical morphology which refers to the study of geometrical structures. Not to be confused with biological cell morphology. Extended to grayscale, *watershed* is a part of this family since it treats images as topographical surfaces. In this section a few basic binary morphological operations will be presented that can easily be extended and defined the grayscale domain.

Morphological operations in image processing are based on images where objects (foreground) are represented as sets that are processed with smaller structuring elements. A structuring element is a small shape represented by a pixel matrix. The following discussion is based on a book on image processing. [28] The operations explained here are the foundation of many complex algorithms in morphological image processing. Example images are provided in figure 2.3-2.6, the original image is 140x125 pixels and a disk-shaped structuring element with radius 5 pixels was used in each image.

Erosion

The definition of erosion in an image containing the foreground pixel set(s) A with the structuring element B is

$$A \ominus B = \{z \mid B_z \subseteq A\} \tag{2.11}$$

where $(B)_z$ denotes the structuring element B translated with z. In other words, for every pixel z, does the translated structuring element B_z fully fit into the original pixel set A? If yes, this pixel is part of the eroded image $A \ominus B$.

The effect of erosion is to reduce the size of objects. How much smaller they become is determined by the size of the structuring element. Any object smaller than the structuring element, such as dots or lines, will disappear.

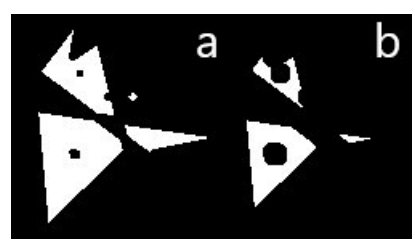

Figure 2.3: Example of a) original image and b) eroded image.

Dilation

Dilation is essentially the opposite of erosion. The effect of dilation is to enlarge the size of objects in an image by a size determined by the structuring element. Using the same notation dilation is defined by

$$A \oplus B = \{z \mid B_z \cap A \subseteq A\}\,. \tag{2.12}$$

In other words, for every pixel z, does the translated structuring element B_z have any overlap with the original pixel set A? If yes, this pixel is part of the dilated image $A \oplus B$.

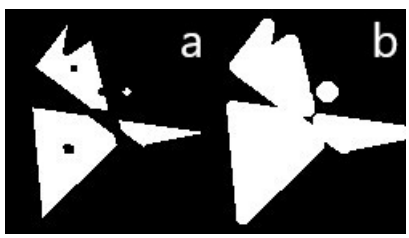

Figure 2.4: Example of a) original image and b) dilated image.

Opening

Opening is erosion followed by dilation with the same structuring element,

$$A \circ B = (A \ominus B) \oplus B. \tag{2.13}$$

The effect of opening is removing small objects and eliminating thin structures such as narrow bridges.

Figure 2.5: Example of a) original image and b) opened image.

Closing

Closing is dilation followed by erosion with the same structuring element,

$$A \bullet B = (A \oplus B) \ominus B. \tag{2.14}$$

The effect of closing is eliminating small holes and fusing narrow breaks.

It is worth noting that the perimeter or overall size of objects do not change when opening or closing.

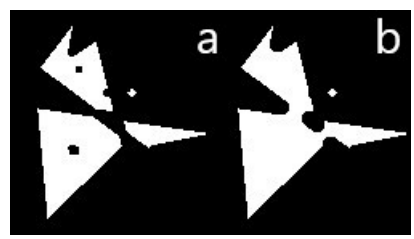

Figure 2.6: Example of a) original image and b) closed image.

2.3.4 Watershed

Watershed is a grayscale morphological image segmentation method that can automatically separate touching objects from each other. A watershed etymologically refers to a geological ridge dividing two water drainage areas. This is essentially what the watershed algorithm seeks to emulate, viewing the image as a topographical surface, as in figure 2.7. An image contains ridges and valleys depending on high or low pixel intensities respectively. There are a few variations of watershed algorithms. This thesis used the MATLAB implementation of watershed which is based on the Fernand Meyer algorithm. [29]

A way to visualize the watershed algorithm is to imagine creating small holes in the surface at the locations of all regional minima. Then slowly starting to submerge the surface into water at constant velocity. The valleys in an image then start to act as catchment basins. Eventually two catchment basins will flood over into each other. At these locations a one-pixel infinitely high border is constructed partitioning the regions. These borders are called watershed ridges. It is important to note that the watershed does not take into account the magnitude of the regional minima. Thus, it will create small catchment basins in the regional minima of noise. This is further addressed below. When every pixel in the image is divided into regions the process is complete and the image

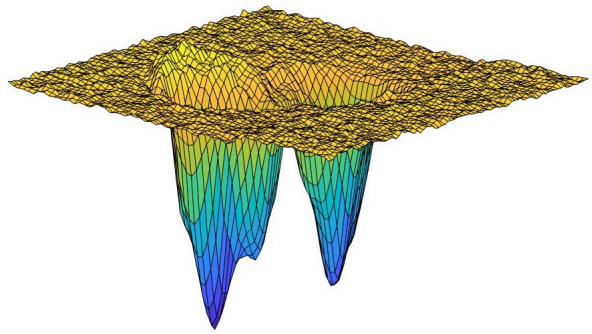

Figure 2.7: Topographic representation of the image in figure 2.1a. The image was inverted to show catchment basins rather than mountains.

has been segmented into regions. The exact implementation of this process can vary, many simply simulating the process of flooding a topographical map. [29]

Binary Watershed

Watershed can be performed on binary images. Before the image can be inputted to the watershed algorithm it must be converted to grayscale with a fitting topographical map. One way to do this is to use the Euclidean distance transform. For every pixel within the mask the Euclidean distance is computed to the nearest pixel outside the mask, see figure 2.8. This creates high values for pixels firmly in the middle of the mask with a high probability of being a true catchment basin.

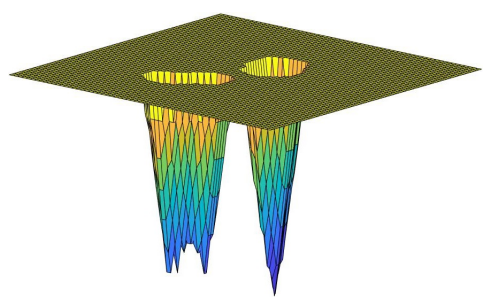

Figure 2.8: Euclidean distance transform of the binary image in figure 2.1b. The image was inverted to show catchment basins rather than mountains.

It is important to note that when the flooding of the pixels reaches the top of figure 2.8 it spills over the whole surface immediately, see figure 2.9a. A mask

is needed to subtract the background from these regions. In figure 2.9b it is clear that the image has been successfully segmented when removing the background using the mask.

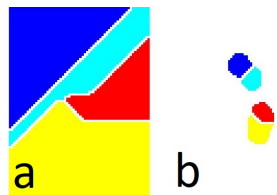

Figure 2.9: a) watershedded image from figure 2.8 and b) the same image with the background removed using the mask in figure 2.1b. In this case the two lobes of the bilobular nuclei of the neutrophils have been segmented. This shows that a binary mask is needed to produce a correct watershed. However, it should be noted that it might not always be desirable to segment the lobes separately and depends on the application.

Grayscale Seeded Watershed

Watersheds can also be directly applied on grayscale images. Typically, this requires some preprocessing in advance. Watershedding an unprocessed image usually results in a phenomenon called oversegmentation, see figure 2.10. This is because the algorithm uses all regional minima in the image regardless of size.

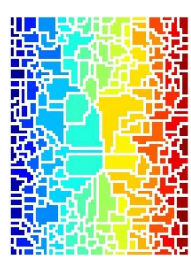

Figure 2.10: Watershed oversegmentation of the image in figure 2.1a due to the lack of preprocessing.

A solution to oversegmentation is providing the algorithm with predetermined starting points, called seeds. The seeds can be obtained in a multitude

of ways, often specific to the specific dataset. When the seeds have been obtained the original image can be morphologically reconstructed such that the seed points become the only regional minima in the image. This process is called minima imposition. The seeds can be thought of as the only holes that are punched in the topological surface before submerging it. The watershed algorithm can then proceed as normal.

In MATLAB, Morphological reconstruction for minima imposition can be performed using a single line of code and a function called `imimposemin`. The details of this algorithm is not presented here. For an interested reader a reference is provided [30], which is the basis for the implementation in MATLAB.

In brief, morphological reconstruction is performed by using a mask image posing restrictions on an original grayscale image and a structuring element. In the case of minima imposition, the mask image contains the locations of the seed points that will become regional minima. The structuring element is neighbourhood of eight pixels around each pixel. At the location of the seeds, the pixel values are statically set to 0. Then, the algorithm iteratively uses grayscale morphological operations until the image conforms to having only regional minima at the locations of the seeds. Please refer to figure 2.11 and 2.12 to visualize the effect of minima imposition. In figure 2.11a, all regional minima of an image are shown, and in 2.11b, one single minima has been imposed by morphological reconstruction. In figure 2.12, the effect on the surface of imposing one single minima is shown.

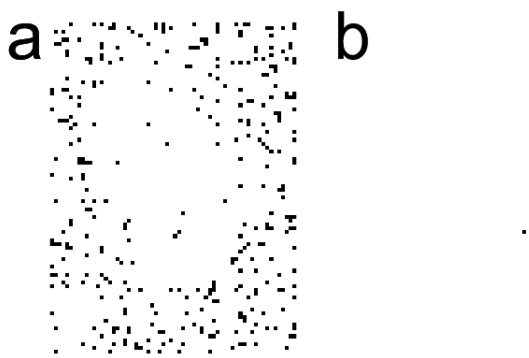

Figure 2.11: a) all regional minima of the surface in figure 2.7 plotted as black dots and b) all regional minima of the same surface after minima imposition using a single seed.

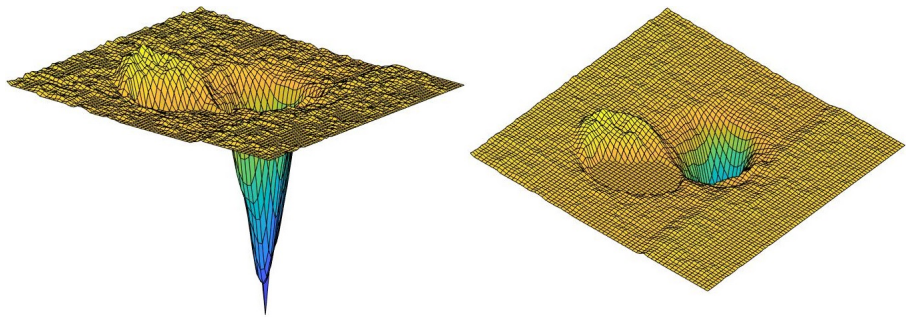

Figure 2.12: One single imposed minimum through morphological reconstruction on the surface in figure 2.7. Side view and top view. Notice that all other regional minima have been suppressed. This is the same image as represented in 2.11b.

2.3.5 Gaussian Filter

A gaussian filter smoothens images and suppresses noise. This is done by convolving an image with a 2-D Gaussian function [28], which from statistics has the well known form

$$G(a,b) = \frac{1}{2\pi\sigma} e^{-\frac{a^2+b^2}{2\sigma^2}}, \tag{2.15}$$

where σ is the standard deviation of the Gaussian distribution. In image processing it is most often implemented symmetrically with respect to the axes, with the change of variables $r = (s^2 + t^2)^{1/2}$. The Gaussian function is then downsampled into discrete pixel values in a small sub image. This image is referred to as a convolution kernel. This kernel is then swept over all pixel positions in the original image. For each position, the sum of the products of all overlapping pixels between the image and the kernel is calculated. This is called image convolution.

The definition of image convolution is

$$[w * f](x,y) = \sum_{s=-a}^{a} \sum_{s=-b}^{b} w(s,t) f(x-s, y-t), \tag{2.16}$$

where $f(x,y)$ is the image and w the kernel of size mxn. m and n are odd integers (to allow for a center pixel) whereas $a = (m-1)/2$ and $b = (n-1)/2$. [28] This is also referred to as a spatial linear filter with w.

A Gaussian kernel with size 5 x 5, $\sigma = 0.5$ computed to four decimal places

looks like

$$\begin{pmatrix} 0.0000 & 0.0000 & 0.0002 & 0.0000 & 0.0000 \\ 0.0000 & 0.0113 & 0.0837 & 0.0113 & 0.0000 \\ 0.0002 & 0.0837 & 0.6187 & 0.0837 & 0.0002 \\ 0.0000 & 0.0113 & 0.0837 & 0.0113 & 0.0000 \\ 0.0000 & 0.0000 & 0.0002 & 0.0000 & 0.0000 \end{pmatrix}. \tag{2.17}$$

This kernel is translated across every pixel in the original image setting the new values to be the weighted average of the pixel vicinity specified by the kernel. The center pixel value is the most influential pixel but values of surrounding pixels also bleed into it. This causes the blurring effect. The larger the σ the larger the blurring effect.

Because the Gaussian function decreases exponentially from the center it is not necessary to compute an unnecessarily large Gaussian kernel. In fact, a $m \times m$ kernel were $m = \text{ceiling}(6\sigma)$ is completely satisfactory. [28] Thus, the outer perimeter of the Gaussian kernel in eq. 2.17 is not computationally necessary.

Chapter 3

Methods

3.1 Data Acquisition

This section describes how the data used in this project was obtained. These steps were carried out before the thesis work began by Cecilia Magnusson. This is provided for the interested reader.

Blood samples were drawn from male patients with confirmed metastasized prostate cancer to vacutainer blood collection tubes containing EDTA (anti-coagulant). Within four hours 6 ml of the blood was treated with BD FACS™lysing solution for 15 minutes in room temperature to lyse red blood cells. This ruptures the membrane of red blood cells. The cells were centrifuged in 400g for 5 minutes and the remaining fluid was removed. The cells were then fixed with 4% paraformaldehyde incubated for 25 min in room temperature.

To wash the cells the mixture was centrifuged in 400g for 5 minutes and the fluid was removed. The cells were dissolved in FACS buffer solution and centrifuged again in 400g for 5 minutes. The fluid was removed and then the cells were dissolved in 12 ml FACS buffer.

To further separate CTCs from WBCs a microfluidic acoustophoresis CTC enrichment chip was used. This method to separate CTCs using sound has been described elsewhere [10] and a principle schematic has been provided in figure 3.1. In brief, cells flow through a microchannel where a first acoustic field aligns

them into two distinct bands. Thereafter, a second acoustic field separates cells based on their acoustic mobility. Cells that are large or have high acoustic contrast tend to migrate to the central outlet, whereas small or low acoustic contrast cells exit through the side outlets. The sample was run at 75 µlmin^{-1} using an acoustic wave that should sort CTCs with a contamination rate of about 2% WBCs.

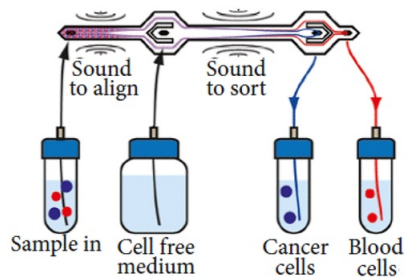

Figure 3.1: Principle schematic of acoustophoresis. [10]

Before immunostaining, the cells were centrifuged in 400g for 5 minutes and the fluid was removed. The surface marker antibodies EpCAM-PE (40 µl), CD45-APC (40 µl) and CD66b-AF (10 µl) were added to the cells along with 70 µl FACS buffer and incubated for 25 minutes in room temperature. CD66b was not used in early patient samples.

Thereafter, 10 ml SAP (saponin) buffer was added. This is done to change the buffer in the sample by diluting the FACS buffer along with loose antibodies without losing too many cells. The SAP buffer is designed to make small holes in cell membranes such that intracellular antibodies can enter. The cells were then centrifuged in 400g for 5 minutes and the fluid was removed. Then the intracellular marker antibody panCK-AF488 (2 µl) with SAP (98 µl) was added and incubated on ice for 1 hour. Then DAPI (1 µl) was added with SAP (899 µl) and incubated on ice for 5 minutes.

10 ml SAP buffer was added, the sample was centrifuged in 400g for 5 minutes and the fluid was removed. The sample was then washed twice by adding 10 ml, the first time with SAP buffer then with FACS buffer, and centrifuging (400g 5 min) and removing the fluid.

The cells were then dissolved in 200 µl FACS buffer and analysed in the imaging flow cytometry system ImageStream according to the manufacturer's manual. The imaging data is then provided as a .rif-file (raw image file). The data is imported to the associated software IDEAS where the images are adjusted with a calibration file created by ImageStream before running the sample. The

data is then exported as a .cif-file (compensated image file). This is the data used in this thesis work.

Three datasets were provided from the on-going study, patient 6, patient 8 and patient 9. They had also already been analysed in the standard software IDEAS. The number of CTC discovered in each sample had been counted according to the criteria presented in section 1.2. A list of these confirmed cancer cells was provided with the data for the thesis work. A few CTC - WBC clusters were also discovered by time-consuming spot-checking. Setting the gate thresholds is done subjectively and was done by qualitatively inspecting immunofluorescence intensity histograms generated by IDEAS.

FACS buffer: 1x PBS with 1% FBS (fetal bovine serum) and 2 mM EDTA.

SAP buffer: 1x PBS with 0.1% Saponin and 0.5% BSA.

3.2 Datasets

The three patient sample datasets basically contain an array of events detected by ImageStream. For every event there are seven associated images. These images are two brightfield channels, one darkfield scatter channel, one DAPI channel, one panCK channel, one EpCAM channel and one CD45/CD66B channel (henceforth referred to as the CD45 channel). The last four are the fluorescent channels. The images are of 16 bit depth and are generally in a size of slightly smaller than 100x100 pixels.

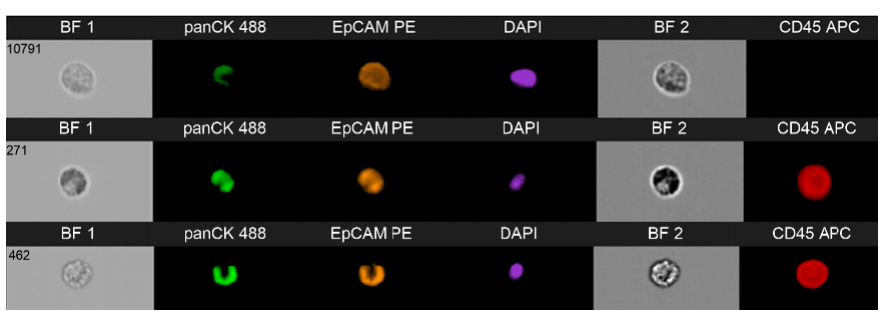

Figure 3.2: Images of three events from the data viewed in IDEAS stacked vertically. The darkfield channel is here omitted and the fluorescent channels have been coloured.

Figure 3.2 shows three examples of events in the data. The top event probably depicts a cancer cell as it has a positive EpCAM and DAPI signal, panCK is debatably positive, and has no CD45 signal. The latter two are probably WBCs

since they have a strong signal in the CD45 channel and a positive DAPI. These WBCs then autofluoresce clearly in the panCK and EpCAM channels.

Patient 6 The patient 6 dataset contained 14632 events. In the dataset there were a total of 12 previously confirmed events containing CTCs. 7 of them were classic CTCs (both EpCAM and panCK positive), 3 of them were EpCAM negative CTCs and 2 of them panCK negative CTCs. An important difference to note about this dataset is that is contains no CD66b staining.

Patient 8 The patient 8 dataset contained 128002 events. In the dataset there were a total of 11 previously confirmed events containing CTCs. All of them were classic CTCs. Two of these CTCs were CTC - WBC clusters found by spot-checking and were the only known occurence of such clusters in the datasets.

Patient 9 The patient 9 dataset contained 6715 events. In the dataset there were a total of 48 previously confirmed events containing CTCs. 22 of them were classic CTCs and 26 of them were EpCAM negative CTCs.

3.3 Algorithm Development

The algorithm to detect CTCs from the dataset was developed in MATLAB. The general approach was that of bottom up and trial & error using the listed criteria of CTCs in section 1.2. The algorithm was implemented sequentially by

1. importing data to MATLAB,
2. performing segmentation and
3. region intensity analysis.

The approach was to perform segmentation on the DAPI channel image. The reason for this is that DAPI only stains the nuclei. Since nuclei exist only within cells it should provide an additional spatial separation of signal to make them easier to segment. The segmentation approach was to use Otsu thresholding and watershed. The reason for this is that they are both common techniques in segmentation of fluorescence images. [26] The approach to measure fluorescence intensity was using the region median pixel value. The final algorithm is presented in section 4.1.

In the region analysis stage threshold values need to be set to distinguish a positive signal from a negative. The total intensity of the fluorescence in the

different datasets varied so much that it was not possible to set these threshold values uniformly. This is illustrated in figure 4.12 to 4.14 in the results chapter. Instead, a calibration procedure was developed to let the user set the threshold values for each dataset. This created a histogram of all segmented regions' median intensities for every channel. Using these histograms it is possible to set reasonable threshold values.

Segmentation

While developing the segmentation algorithm, continuous evaluation of the visual effect on the DAPI image was performed. In order to finally assess the effectiveness of the segmentation method a test was performed. In IDEAS, a population of events was extracted where only cell clusters or larger pieces of debris were present. This was done by selecting events with height and width larger than that of a typical single cell in the standard DAPI mask. It was not needed to test segmentation on single cells. The population was extracted from the patient 8 dataset. The segmentations of the first 200 events in this population (events not containing any DAPI signal were skipped) was examined visually and compared to respective DAPI image. They were deemed either perfect, acceptable or unsatisfactory. The development of regional intensity analysis did not continue until satisfactory segmentation results were attained.

CTC Detection in Datasets

When the algorithm was complete in full, it was tested on a subset of the previously confirmed CTCs by Cecilia Magnusson from patient 8. If a confirmed CTC was not detected, the algorithm was debugged and algorithm parameters were changed accordingly to correct the underlying problem.

Finally, to test the algorithm, a CTC detection test was performed on the provided datasets. The datasets pertaining to patient 6, 8 and 9 were run through the algorithm. It was noted whether the algorithm could detect all previously confirmed CTCs and how many possible CTCs were detected. All detected possible CTCs were also assessed manually by Cecila Magnusson. They were grouped into two categories, either interesting or false-positive.

There were unfortunately not enough examples of known CTC - WBC clusters at the time of this thesis to conduct a proper quantitative test on whether the algorithm could detect them in particular.

GUI

As a final step, a graphical user interface (GUI) was developed through which the algorithm could be applied to datasets. This was created as a MATLAB app. In addition, another linked MATLAB app was developed where the results could be visualised and sorted into classes defined by the end user. These classes were classic CTC, EpCAM+/panCK- CTC, EpCAM-/panCK+ CTC and debris, as well as distinguishing clusters from single cells.

Chapter 4

Results

4.1 Overview of Algorithm

This section explains the proposed algorithm solution in detail based on figure 4.1. The data arrives as a compensated image file (.cif). The Bio-Formats library was used to load the data into MATLAB. [31] Bio-Formats is an open-source library widely used to load life science image files. A wrapper function was constructed to facilitate communication between MATLAB, Bio-Formats and the data in question.

The DAPI channel image is used for cell segmentation. Firstly, images not containing a DAPI signal is skipped. This is done by comparing the maximum pixel intensity to a predetermined value. The value used here was the same as the DAPI threshold used in the analysis later. Please refer to the section 4.1.1 on how this value is set. The main reason for this is to minimize computational expense by skipping following image processing steps.

Shown in the left branch of figure 4.1, a binary segmentation mask is obtained, determining where there is signal. To do this an Otsu threshold is applied to the image returning a binary image representing foreground (1) and background (0). Thereafter small objects are filtered, any 8-connected region with fewer pixels than a specific value is removed. This value was set to 70 pixels, which was later shown to be the largest value that did not miss any confirmed

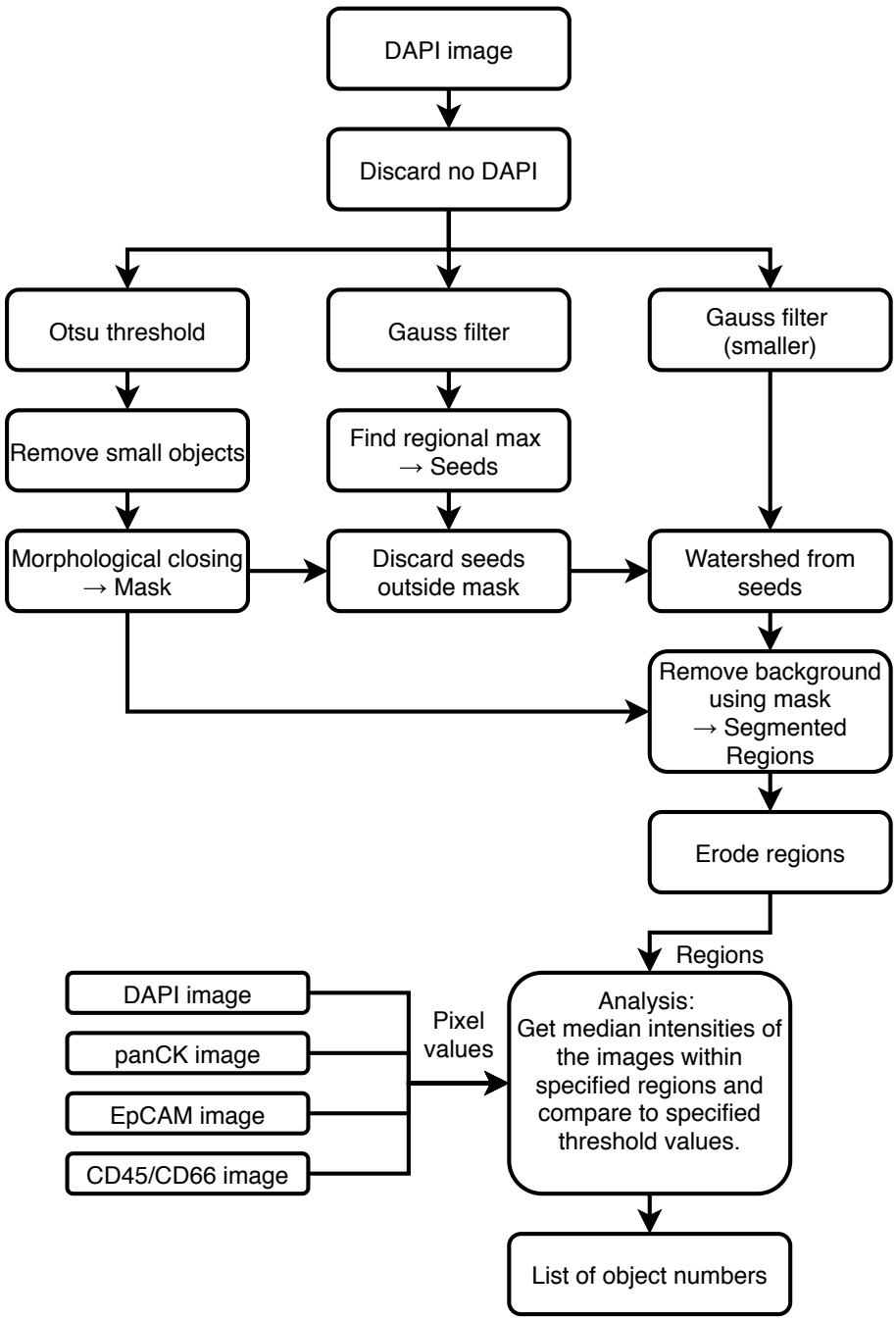

Figure 4.1: Flowchart overview of algorithm, which is explained in detail in this section.

cancer cells in the datasets. To fuse holes present inside a mask morphological closing is applied using a square 5x5 shaped structural element. This binary image is referred to as the mask.

Shown in the middle branch of figure 4.1, seeds are obtained to be used as starting points for the watershed algorithm. To smoothen the DAPI image and avoid more than one local maxima per nucleus, it is subject to a Gaussian filter. The Gaussian smoothing kernel is determined by the standard deviation. This value was set to 4 and was set by visually looking at the final seed points on subsets of the data. Increasing this value joins close seeds and leads to fewer seeds being found. Regional maxima in the smoothed image is then detected. A maximum, or seed point, is defined as a pixel where all its 8-connected neighbours are less than or equal to said pixel. Connected maxima are treated as one seed point. Thereafter, any seeds existing outside the mask is discarded.

Shown in the right branch of figure 4.1, another gaussian filter is applied to the DAPI image. The standard deviation was here set to 0.5 by visually inspecting the segmentation results on a subset of the data. The resulting image is then used in watershedding. Before watershedding, the image is inverted, and using morphological reconstruction the seed points are manipulated to be the only regional minima in the image. This is done using MATLAB's `imimposemin`-function, as discussed in section 2.3.4. When watershedding the catchment basins originate from the seeds. After watershedding the mask is used to remove the background. The results are labelled regions where different nuclei have been separated. The regions are then slightly morphologically eroded using a 3x3 matrix structural element where the corners are set to zero. This is to decrease the chance of overlapping cells bleeding signal to each other in the other channels.

The labelled regions are then used to determine if there is a potential CTC at this location. The user can decide whether to only analyse images with more than one region, i.e. clusters, at this point. The analysis was made according to the criteria established in section 1.2. For each segmented region, the median pixel intensities are calculated for the DAPI, panCK, EpCAM and CD45 channels. These values are compared to predetermined threshold values, please refer to the calibration procedure, section 4.1.1, on how the thresholds can be set. If a region is DAPI positive, CD45 negative and either panCK or EpCAM positive it is marked as potential cancer cells. After scanning the whole dataset, a list of image numbers is provided to the user containing potential cancer cells.

4.1.1 Determining Thresholds

The issue of determining thresholds for the DAPI, panCK, EpCAM and CD45 channels was addressed by creating a way to visualize the data. Firstly, images not containing a DAPI signal are skipped. Otherwise, this could yield strange results as the segmentation could be performed on pure noise. This is done by comparing the maximum pixel intensity to a value which is set by the user. A value about three times larger than the background worked well on the provided datasets.

The rest of the algorithm is then performed exactly as previously described. However, instead of outputting a list of object numbers, the median intensities of each channel are saved in arrays. Then the intensities of each channel are plotted in histograms. The histograms of the provided datasets can be viewed in figure 4.12 to 4.14. If the immunofluorescent dye is perfect, two separate peaks should appear in the histograms. The first pertaining to negative signal and the second to positive signal. However, it was noted that the histograms looked very different depending on which patient's dataset was used.

The user needs to qualitatively study these histograms and determine suitable threshold values. This is somewhat subjective in nature. However, if there already is a subpopulation of known cancer cells the user can use only these in the calibration algorithm. Then, find thresholds that make sure to not miss any of them. This would make a good estimate for the rest of the dataset. This would be ideal if the previous IDEAS method already has been employed and this program is only used to locate CTC-WBC clusters.

4.2 Examples of Detected Events

Figures 4.2 to 4.9 show examples of detected events containing CTC candidates. The examples were taken from the patient 8 dataset and were all marked as clusters by the algorithm.

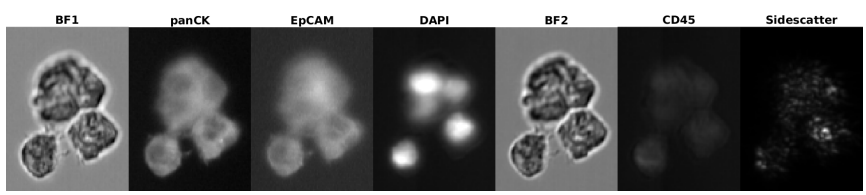

Figure 4.2: Five clustered CTCs. Previously confirmed by Cecilia Magnusson.

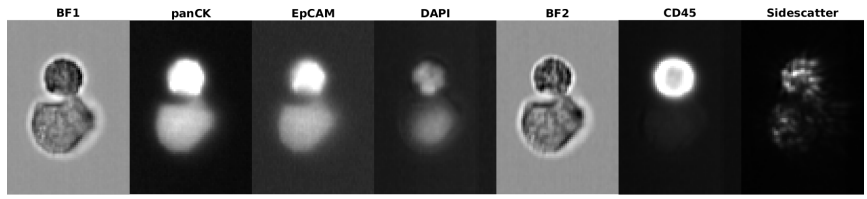

Figure 4.3: CTC-WBC cluster. The CTC, the bottom cell in the image, was previously confirmed by Cecilia Magnusson by spot-checking.

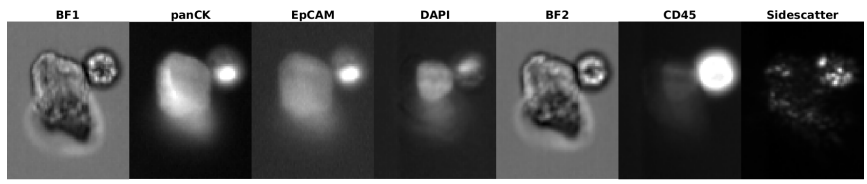

Figure 4.4: Previously unknown CTC-WBC cluster. There is one, perhaps two, CTCs to the left of a WBC.

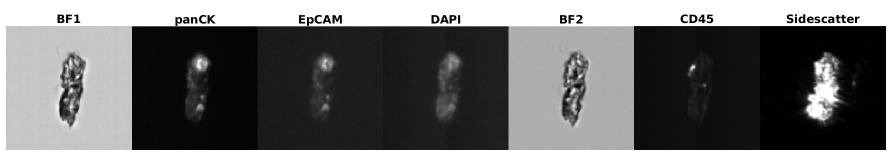

Figure 4.5: This was marked as false-positive. It has fragmented DAPI and odd-looking fluorescence.

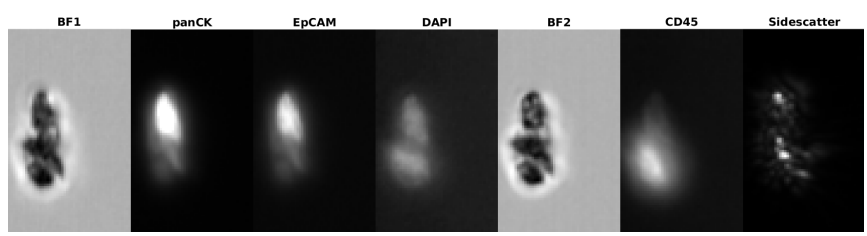

Figure 4.6: The top object could be a CTC. However, it does not look like a cell in the brightfield images. This was marked as interesting.

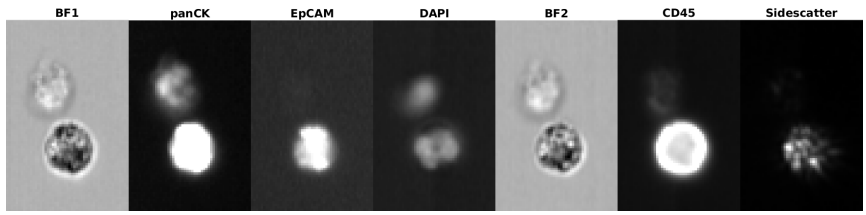

Figure 4.7: Possibly a CTC-WBC cluster, where the CTC is EpCAM negative. This was marked as interesting.

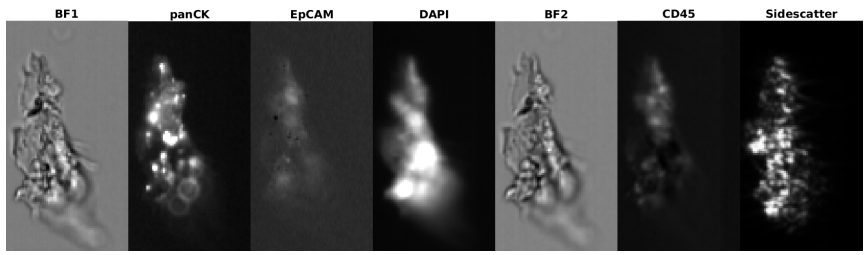

Figure 4.8: This is a false-positive piece of debris. Auto-fluorescence in the DAPI image caused over-segmentation and eventually a CTC detection.

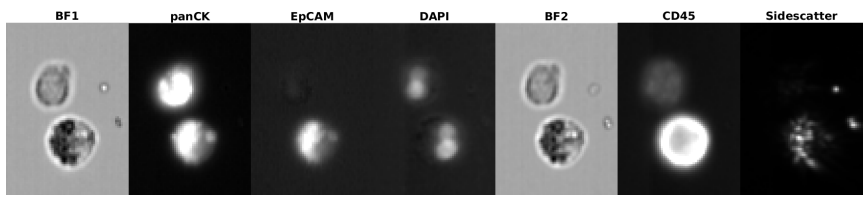

Figure 4.9: False-positive. The top cell has fragmented DAPI. Noteworthy, the CD45 signal is low.

4.3 Segmentation

The image segmentation of cells was performed on the nuclear staining channel DAPI and was evaluated on 200 images. A few examples of the results of this can be seen in figure 4.10. Out of the 200 images all were determined to be acceptable segmentations, where 186 were deemed as perfect segmentations. In general, the non-perfect segmentations sprung from providing too few seeds. This can be somewhat adjusted using a smaller standard deviation in the Gaussian filter.

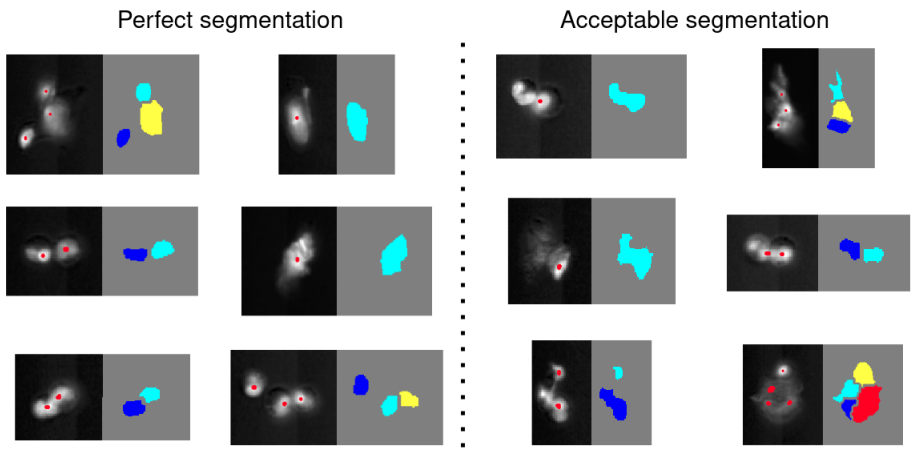

Figure 4.10: Twelve examples of the final seeded watershed segmentation method. Original DAPI images to the right and coloured segmented regions to the left. The seeds are marked in red.

Binary Watershed Segmentation

Another test was conducted on an alternative watershed algorithm and is provided here as means of comparison. This is the binary watershed segmentation method presented in section 2.3.4. This method was not used in the final algorithm. A few examples of the results of this can be seen in figure 4.11. This algorithm worked quite well but performed weakly specially on complex nuclear structures.

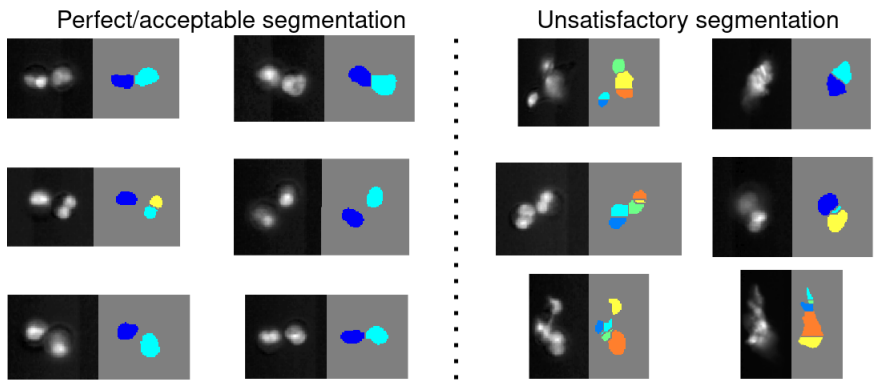

Figure 4.11: Twelve examples of the binary watershed segmentation method on DAPI images. Original DAPI images to the right and coloured segmented regions to the left.

4.4 Detection of CTCs from Datasets

In this section, the initial test results on the three datasets are presented. In all three patient samples every previously confirmed CTC by Cecilia Magnusson was detected.

The calibration histograms are presented in figure 4.12 to 4.14 and the thresholds are marked. Ideally, the thresholds should be between two peaks, the first corresponding to negative signal and the second to positive. However, this was not always the case. It is important to note that in each individual histogram there are many positive (or negative in the case of CD45) events where CTCs, WBCs and debris are mixed. However, CTC detection only occurs when one single region is DAPI positive, panCK or EpCAM positive and CD45 negative.

4.4.1 Patient 6

The calibration histograms of patient 6 are shown in figure 4.12. The sample included 14632 events. This patient was not stained in with CD66b and thus the CD45-histogram has an extra peak at about 200 intensity units which corresponds to the granulocytes, that do not express CD45. The threshold values were chosen by looking at the intensity values of previously confirmed CTCs and confirming the plausibility the histograms of figure 4.12. The values chosen were: DAPI 170, panCK 100, EpCAM 100 and CD45 110 intensity units.

Using these threshold values the algorithm detected 104 events containing possible CTCs. Every previously confirmed CTC event was detected. Out of the remaining 92 events 9 were deemed interesting, the rest were false-positive. The false positive rate was thus in this case 79.8 %. By choosing to only include events with more than 2 segmented regions, i.e. clusters, less than ten possible events containing CTCs were identified.

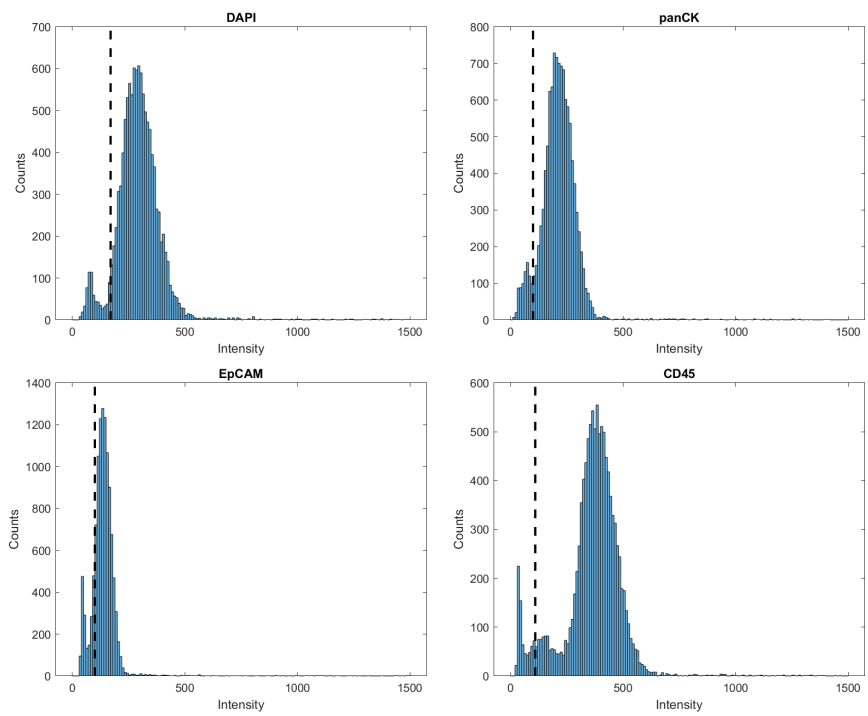

Figure 4.12: Patient 6 calibration. The used thresholds have been marked.

4.4.2 Patient 8

The calibration histograms of patient 8 are shown in figure 4.13. The sample included 128002 events. The threshold values were chosen by looking at the intensity values of previously confirmed CTCs and confirming the plausibility the histograms of figure 4.13. The values chosen were: DAPI 100, panCK 100, EpCAM 100 and CD45 150 intensity units. Compared to other patient samples the DAPI signal was low.

Using these threshold values the algorithm detected 324 events containing possible CTCs. The amount of detected events were too high to manually assess. However, every previously confirmed CTC event was detected. By choosing to only include events with more than 2 segmented regions, i.e. clusters, only 23 possible events containing CTCs were identified. Out of these, one previously unknown CTC - WBC cluster was detected, which is the only new one detected by this algorithm.

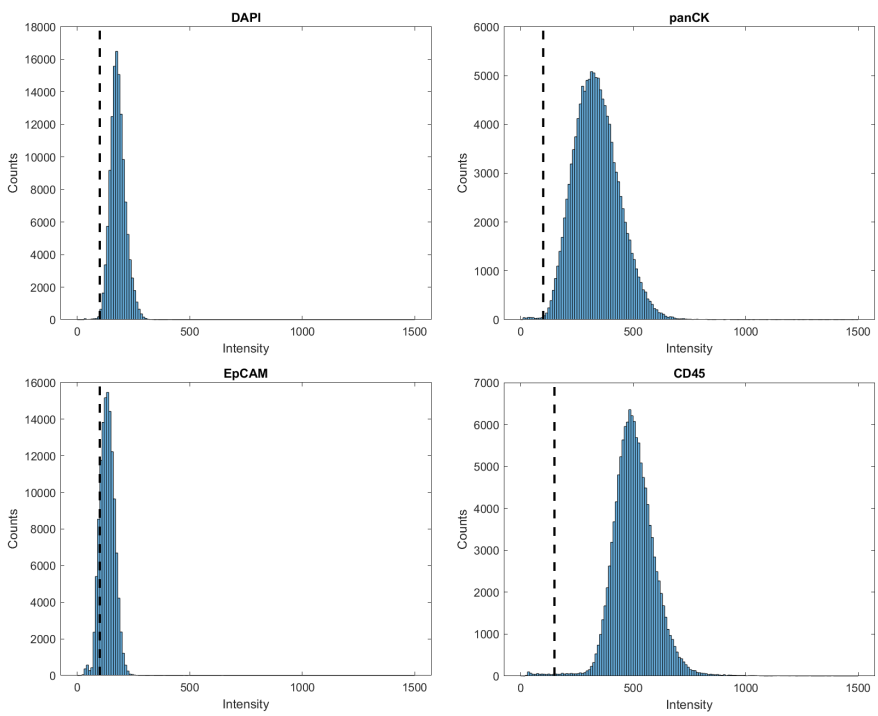

Figure 4.13: Patient 8 calibration. The used thresholds have been marked.

4.4.3 Patient 9

The calibration histograms of patient 9 are shown in figure 4.14. The sample included 6715 events. The threshold values were chosen by looking at the intensity values of previously confirmed CTCs and confirming the plausibility the histograms of figure 4.12. The values chosen were: DAPI 120, panCK 100, EpCAM 100 and CD45 150 intensity units.

Using these threshold values the algorithm detected 108 events containing possible CTCs. Every previously confirmed CTC event was detected. Out of the remaining 60 events 14 were deemed interesting, the rest were false-positive. The false positive rate was thus in this case 42.6 %. By choosing to only include events with more than 2 segmented regions, i.e. clusters, 17 possible events containing CTCs were identified.

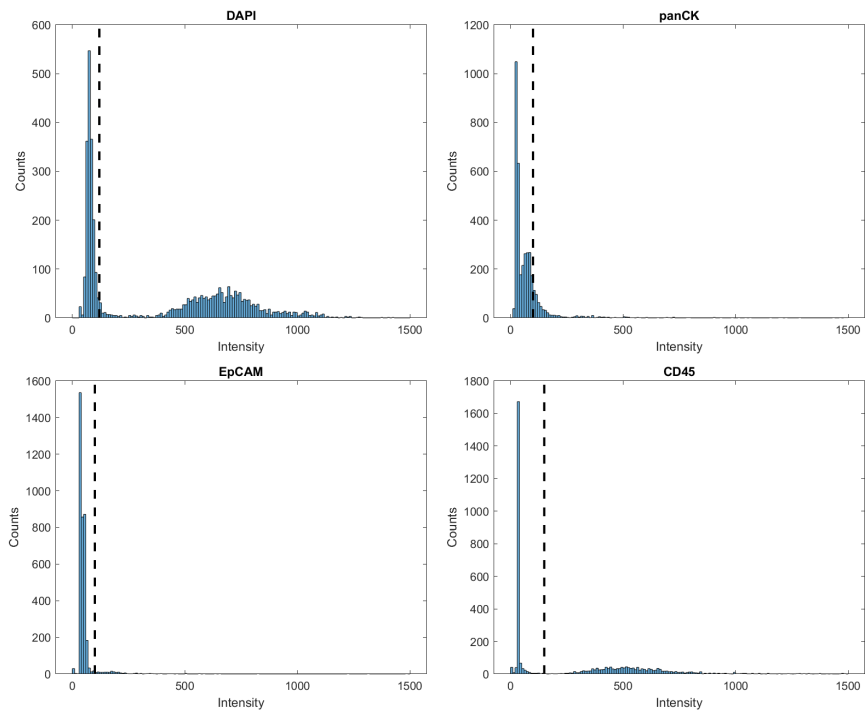

Figure 4.14: Patient 9 calibration. The used thresholds have been marked.

4.5 Graphical User Interface

The algorithm was made accessible through a GUI. An image of the GUI is presented in figure 4.15. From the GUI it is possible to load a data file, create calibration histograms, enter threshold values and search the data for CTCs. The results are presented as event numbers. Further, the user can plot the images and adjust the contrast of them. The user can also create an IDEAS population file to automatically create an IDEAS population. This is essentially a text file containing the list of event numbers. The user can alternatively launch the sorting app, in which it is possible to step through the data and sort them into classes. An image of the sorting app is presented in figure 4.16.

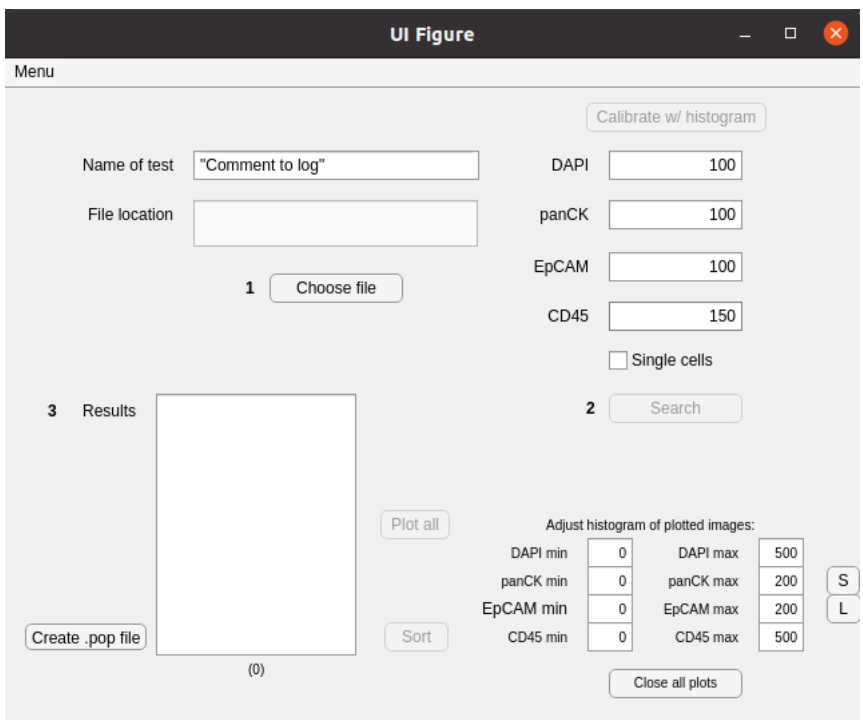

Figure 4.15: Graphical user interface.

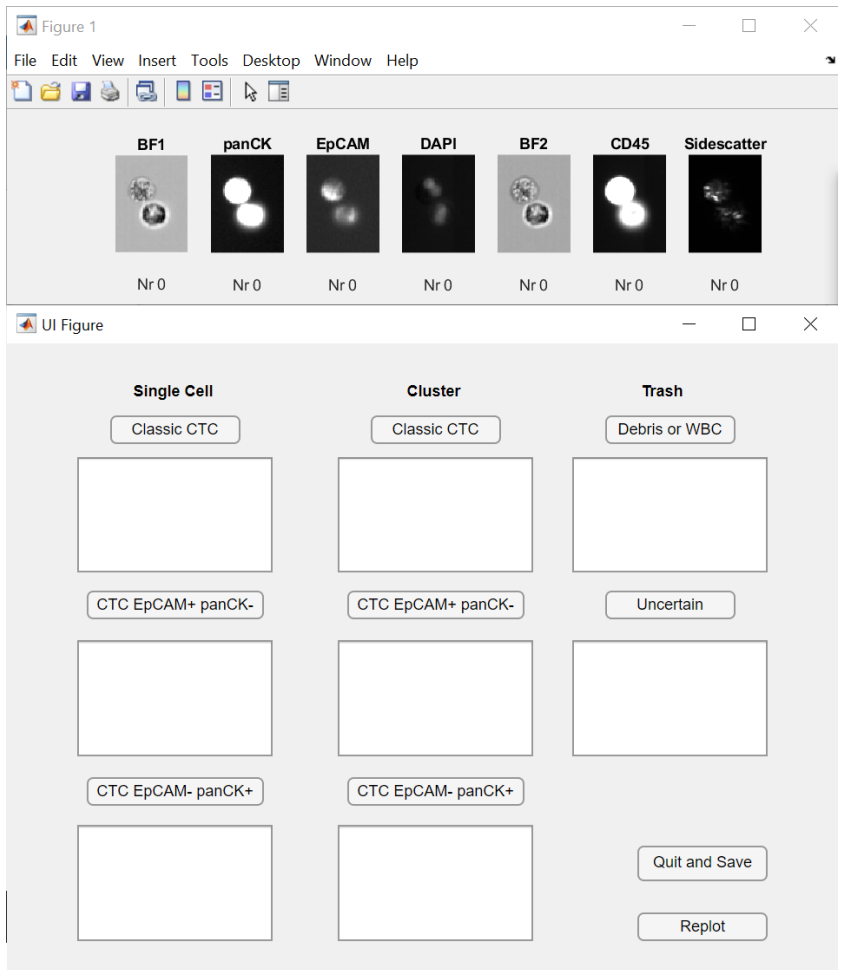

Figure 4.16: Sorting app interface.

Chapter 5

Discussion

5.1 Segmentation

In general, biological data is infamous for low signal-to-noise ratios and biological images are often hard to segment due to inherent noisiness. The provided imaging flow cytometry datasets, however, proved to provide stable segmentations using the proposed algorithm. For this thesis work, the segmentation performance was determined satisfactory. In this section, some limitations and observations of the proposed method are discussed.

In the thesis, the segmentation algorithm was only performed on the DAPI channel. This discards some of the morphological information of the other channels in regions not overlapping with the DAPI region. It is, of course, possible to use the same segmentation method on the other fluorescence channels. However, since these stains are not as localized as the DAPI nucleus stain, one cannot as easily predict the image appearance in any given scenario. For example, two or more cells close to each other might appear to look like one, etc. One way to avoid this is to use the DAPI seeds as starting points for watershed in the other channels. However, this might cause nearby auto-fluorescing pieces of debris or nuclei-lacking cells to be included in the region. These considerations were only reflected on conceptually and were not put to true test in the thesis.

The main limitation of choosing to only segment in the DAPI channel is

that it will not separate clusters that appear to only have one nucleus, specifically clusters with only one DAPI regional maximum. This can for example happen when a cluster is vertically aligned on the camera axis. For an example of this see figure 5.1, in the algorithm this was detected as one region. In the DAPI channel it appears that there is only one nucleus but if the brightfield image is examined it is clear that there are two cells in question. One cell is slightly superimposed on top of the other. If this was a CTC cluster it would only be detected as a single cell. Also, if this was a CTC - WBC cluster it might not be detected at all depending on the signal intensity and area ratio between them. Noteworthy, this particular case was correctly segmented by the binary watershed approach.

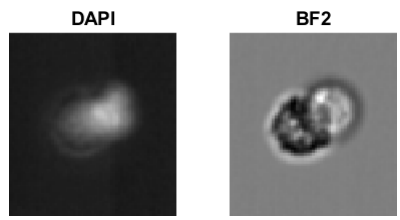

Figure 5.1: A superimposed cell cluster causing the DAPI image to appear to contain a single cell. This presents a challenge to the proposed algorithm.

How finely the regions are segmented is determined by how many regional maxima are detected. This is in turn determined by the Gaussian smoothing kernel before that step. Changing the value of Gaussian standard deviation here lets one control how small segmentations regions are needed. For example, if one wants to segment the two lobes of a neutrophil nucleus, such as in figure 2.1, one could simply lower the standard deviation. This could be used for detection of neutrophils when cross-examined with the original segmentation. However, this could cause unwanted oversegmentation in other instances. In this thesis, a standard deviation was chosen such that neutrophil nuclei were segmented in one region, this meant it would not be detected as a cluster to the user.

5.2 CTC Detection

The aim of this thesis has been fulfilled. The number of detected clusters to be manually assessed by the user is now acceptable according to Cecilia Magnusson. For patient 6, 14632 events were reduced to 104 or less than 10 clusters. For patient 8, 128002 events were reduced to 324 or only 23 clusters. For patient 9, 6715 events were reduced to 108 or only 17 clusters.

The algorithm managed to detect all previously confirmed CTCs, as well as one previously unknown CTC - WBC cluster. It is, however, not quantitatively proven that it has found all CTCs or will find all CTCs in other datasets. It would not be feasible to go through the datasets manually to find any false negatives and it is possible that some have not been detected. Although, this possibility is assumed to be low. The fact that it has detected one new CTC - WBC cluster and the rarity of these is a good indication that the algorithm works as intended. In this section, a few observations on CTC detection are further discussed.

A major problem when verifying that the algorithm worked in detection of CTCs was the lack of data. Only 71 events known to contain CTCs were available at the beginning of this thesis. Even fewer were CTC clusters, not to mention that there were only two known CTC - WBC clusters. This was a major problem and hindrance for the thesis work. If there were more datasets the algorithm could have been tested more properly and the algorithm could have been more finely tuned. As it stands, the algorithm should be able to detect most CTCs. When more datasets become available through the study in the future, they can be used for improving the algorithm.

When Cecilia Magnusson assessed the results of the CTC detection test she applied more CTC criteria than the definition presented in this thesis. The morphology and the shape of the cells in all channels were assessed before marking them as either interesting or false positive. This meant for example that cells with fragmented DAPI were deemed false positives. The proposed algorithm does not discriminate on shape and could cause the algorithm to seem to have lower quality than what it is designed for. This is especially apparent in patient 6 that does not have CD66b staining causing a lot of granulocytes to appear as false positives. This does, however, give an indication on potential for improvement in the future. Perhaps one could create a shape discriminant in addition to this algorithm, this is discussed further in section 5.4.

5.2.1 False Positives

The main drawback of the proposed algorithm is the amount of detected false positives. The main culprit of this is auto-fluorescence and the fact that debris and WBCs tend to express it substantially. The algorithm cannot distinguish between an actual CTC and a piece of debris that exhibits DAPI and a cancer marker without CD45. This could be improved upon and is discussed further in section 5.4.

Patient 6 The sample from patient 6 was not stained with CD66b. This meant some granulocytes exhibited low CD45 signal (as discussed in section 2.2.4) and made them hard to distinguish from CTCs, see the extra peak in 4.12. Many false positives originated from this and were discarded because of fragmented nuclei. It is apparent that CD66b staining is important in detecting CTCs using this method.

Patient 8 The sample from patient 8 seemed to exhibit unusually low levels of DAPI staining, see figure 4.13. The peak of DAPI positives in patient 8 is at around 160 intensity units while the peak in patient 6 and 9 were around 300 and 700 respectively. This probably caused debris to have comparable levels of DAPI signal which creates a lot of false positives by debris. Although patient 8 had as many as 324 positive events, it should be noted that the size of the dataset was an order of magnitude larger than the other two, which is another reason for the many positive events. It is apparent that proper DAPI staining is important in detecting CTCs using this method.

Patient 9 The patient 9 sample was in many ways the ideal sample where the false positive rate was only 42.6 %. If the upcoming datasets will be similar to this the proposed algorithm will work well.

5.2.2 Determining Thresholds

Another drawback of the method is that it is not entirely automatic, it needs the user to determine threshold values. It was briefly investigated whether an automatic threshold setting algorithm could be constructed. An Otsu threshold approach and a standard deviation of the histograms approach were briefly tested but did not offer anticipated results. A major hindrance in finding a good threshold setting algorithm was that the dataset histograms, see figure 4.12 to 4.14, were considerably different. Perhaps this can be more easily implemented when more datasets are available.

5.3 Recommended Use of Algorithm

There are two recommended ways to use the developed program in the study. The first is to completely replace the previous software, and the second, perhaps more realistic, is to use it as a complement. The initial tests in this thesis were made according to the second way.

1) Choose the data file in the program. Press "Calibrate w/ histogram" to produce the dataset histograms and select reasonable threshold values. Replace the standard values with these selected values.

In general, the histograms should have two peaks, the first corresponding to negative signal and the second to positive signal. The thresholds should be set such that they allow for the detection of events conforming to the CTC criteria in section 1.2. One can adjust the threshold settings in order to be more conservative or inclusive on the amount of detected events. The more inclusive the thresholds are set the more events are detected. Conversely, the more conservative the thresholds are set the risks of failing to detect CTCs are higher.

Then, tick the "single cells" box to include all detected cells and search the dataset. The results can then be examined in the sorting app. The interface comes with the option to adjust the contrast of the channel images individually.

2) Conduct a normal CTC detection in IDEAS. When a CTC population has been created it can be extracted as a single .cif file. Choose this data file in the program and then press "Calibrate w/ histogram". The plotted values now only represent confirmed CTCs. Thus, it is straightforward to set threshold values. Replace the standard values with these new values. Then choose the original data file instead and search the dataset. In this case the "single cells" box does not need to be ticked as all single cells should already have been discovered in IDEAS. This means fewer detected events will be presented. When the results are printed press "Create .pop file" in order to get a file that can export the results back to IDEAS.

If no CTCs are detected in IDEAS, this particular approach cannot be used. In such a case, refer to the first recommended use of the algorithm.

5.4 Future work

In this section a few ideas to improve upon the work are presented. In addition to these ideas, it would be advisable to test the algorithm continually as more datasets become available to avoid missing CTCs due to some unforeseen issue.

The main reason for false positives being detected in the algorithm is that some debris auto-fluoresce in the CTC signatures. In the current algorithm, this is not addressed. Naturally, an improvement would be to find a way to detect the debris. One way to do this could be to use machine learning. However, this is probably only possible as more data becomes available and labelled. Perhaps one could create a simple model that classifies objects in the brightfield images as cells or non-cells. Convolutional neural networks could be a natural first approach. It would also be possible to try to distinguish fragmented or non-round nuclei, which do not appear in CTCs. Deep learning on imaging flow cytometry data is already widely discussed in the scientific community. [11, 32]

Currently, a lot of information is not used by the algorithm. Perhaps one could try segmentation in the other channels. This might make it possible to find new relationships in morphology and perhaps one could find some superimposed clusters. It is also possible to extract more features from the already existing DAPI segmentation. Please refer to MATLAB documentation on the function `regionprops` for convenient to use features. [33] When extracting more features, the complexity in visualization will increase. One suggestion to visualize the data is the package Voyager 2, available on GitHub. [34] It is also possible to conduct machine learning on features. For example, using random forest classifiers as described in [35].

In the future, it would be interesting to look at automatic threshold setting. When this imaging flow cytometry based CTC detection method matures, it might be easier to automatically set the threshold values. Perhaps by using Otsu thresholding.

It would also be interesting to produce synthetic data to test the algorithm on. A major problem in the thesis was the lack of known true positive CTC - WBC clusters. If one could produce these artificial images with some similarity to natural images it would be possible to quantitatively test the algorithm. For example, the segmentation of clusters aligned on the camera axis, i.e. superimposed clusters, could be tested. These occurrences are hard to detect and test at present. It may even turn out that the imaging hardware is insufficient in detecting these event.

There is also an open source program similar to IDEAS called CellProfiler. This software was not tested during the thesis, however, it would be interesting to do so. The CellProfiler program can be accessed at *cellprofiler.org*.

Chapter 6
Conclusion

In conclusion, an algorithm to detect CTC - WBC clusters in imaging flow cytometry data was proposed and was initially tested on three datasets. The problem of segmenting cells was mainly solved by Otsu thresholding and watershedding. At the time of writing this, the program is deployed and ready for use in the study. The number of images to manually assess to find CTC - WBC clusters was greatly reduced and is now at a manageable level. The method has not yet been quantitatively proven to find all clusters, however, initial tests on the datasets show that the method seems promising.

References

[1] K. Pantel and M. R. Speicher, "The biology of circulating tumor cells," *Oncogene*, vol. 35, pp. 1216–1224, June 2015.

[2] K. Stoletov, L. Willetts, R. J. Paproski, D. J. Bond, S. Raha, J. Jovel, B. Adam, A. E. Robertson, F. Wong, E. Woolner, D. L. Sosnowski, T. A. Bismar, G. K.-S. Wong, A. Zijlstra, and J. D. Lewis, "Quantitative in vivo whole genome motility screen reveals novel therapeutic targets to block cancer metastasis," *Nature Communications*, vol. 9, June 2018.

[3] M. Cristofanilli, G. T. Budd, M. J. Ellis, A. Stopeck, J. Matera, M. C. Miller, J. M. Reuben, G. V. Doyle, W. J. Allard, L. W. Terstappen, and D. F. Hayes, "Circulating tumor cells, disease progression, and survival in metastatic breast cancer," *New England Journal of Medicine*, vol. 351, no. 8, pp. 781–791, 2004.

[4] A. D. Rhim, E. T. Mirek, N. M. Aiello, A. Maitra, J. M. Bailey, F. McAllister, M. Reichert, G. L. Beatty, A. K. Rustgi, R. H. Vonderheide, S. D. Leach, and B. Z. Stanger, "EMT and dissemination precede pancreatic tumor formation," *Cell*, vol. 148, pp. 349–361, Jan. 2012.

[5] L. Wang, P. Balasubramanian, A. P. Chen, S. Kummar, Y. A. Evrard, and R. J. Kinders, "Promise and limits of the CellSearch platform for evaluating pharmacodynamics in circulating tumor cells," *Seminars in Oncology*, vol. 43, pp. 464–475, Aug. 2016.

[6] L. Millner, M. Linder, and R. Valdes, "Circulating tumor cells: A review of present methods and the need to identify heterogeneous phenotypes," *Annals of clinical and laboratory science*, vol. 43, pp. 295–304, 07 2013.

[7] N. S. Barteneva, E. Fasler-Kan, and I. A. Vorobjev, "Imaging flow cytometry," *Journal of Histochemistry & Cytochemistry*, vol. 60, pp. 723–733, June 2012.

[8] C. Costa, L. Muinelo-Romay, V. Cebey-López, T. Pereira-Veiga, I. Martínez-Pena, M. Abreu, A. Abalo, R. M. Lago-Lestón, C. Abuín, P. Palacios, J. Cueva, R. Piñeiro, and R. López-López, "Analysis of a real-world cohort of metastatic breast cancer patients shows circulating tumor cell clusters (CTC-clusters) as predictors of patient outcomes," *Cancers*, vol. 12, p. 1111, Apr. 2020.

[9] B. M. Szczerba, F. Castro-Giner, M. Vetter, I. Krol, S. Gkountela, J. Landin, M. C. Scheidmann, C. Donato, R. Scherrer, J. Singer, C. Beisel, C. Kurzeder, V. Heinzelmann-Schwarz, C. Rochlitz, W. P. Weber, N. Beerenwinkel, and N. Aceto, "Neutrophils escort circulating tumour cells to enable cell cycle progression," *Nature*, vol. 566, no. 7745, pp. 553–557, 2019.

[10] C. Magnusson, P. Augustsson, A. Lenshof, Y. Ceder, T. Laurell, and H. Lilja, "Clinical-scale cell-surface-marker independent acoustic microfluidic enrichment of tumor cells from blood," *Analytical Chemistry*, vol. 89, pp. 11954–11961, Nov. 2017.

[11] M. Doan, I. Vorobjev, P. Rees, A. Filby, O. Wolkenhauer, A. E. Goldfeld, J. Lieberman, N. Barteneva, A. E. Carpenter, and H. Hennig, "Diagnostic potential of imaging flow cytometry," *Trends in Biotechnology*, vol. 36, pp. 649–652, July 2018.

[12] A. Cossarizza, H.-D. Chang, A. Radbruch, M. Akdis, I. Andrä, F. Annunziato, P. Bacher, V. Barnaba, L. Battistini, W. M. Bauer, S. Baumgart, B. Becher, W. Beisker, C. Berek, A. Blanco, G. Borsellino, P. E. Boulais, R. R. Brinkman, M. Büscher, D. H. Busch, T. P. Bushnell, X. Cao, A. Cavani, P. K. Chattopadhyay, Q. Cheng, S. Chow, M. Clerici, A. Cooke, A. Cosma, L. Cosmi, A. Cumano, V. D. Dang, D. Davies, S. D. Biasi, G. D. Zotto, S. D. Bella, P. Dellabona, G. Deniz, M. Dessing, A. Diefenbach, J. D. Santo, F. Dieli, A. Dolf, V. S. Donnenberg, T. Dörner, G. R. A. Ehrhardt, E. Endl, P. Engel, B. Engelhardt, C. Esser, B. Everts, A. Dreher, C. S. Falk, T. A. Fehniger, A. Filby, S. Fillatreau, M. Follo, I. Förster, J. Foster, G. A. Foulds, P. S. Frenette, D. Galbraith, N. Garbi, M. D. García-Godoy, J. Geginat,

K. Ghoreschi, L. Gibellini, C. Goettlinger, C. S. Goodyear, A. Gori, J. Grogan, M. Gross, A. Grützkau, D. Grummitt, J. Hahn, Q. Hammer, A. E. Hauser, D. L. Haviland, D. Hedley, G. Herrera, M. Herrmann, F. Hiepe, T. Holland, P. Hombrink, J. P. Houston, B. F. Hoyer, B. Huang, C. A. Hunter, A. Iannone, H.-M. Jäck, B. Jávega, S. Jonjic, K. Juelke, S. Jung, T. Kaiser, T. Kalina, B. Keller, S. Khan, D. Kienhöfer, T. Kroneis, D. Kunkel, C. Kurts, P. Kvistborg, J. Lannigan, O. Lantz, A. Larbi, S. LeibundGut-Landmann, M. D. Leipold, M. K. Levings, V. Litwin, Y. Liu, M. Lohoff, G. Lombardi, L. Lopez, A. Lovett-Racke, E. Lubberts, B. Ludewig, E. Lugli, H. T. Maecker, G. Martrus, G. Matarese, C. Maueröder, M. McGrath, I. McInnes, H. E. Mei, F. Melchers, S. Melzer, D. Mielenz, K. Mills, D. Mirrer, J. Mjösberg, J. Moore, B. Moran, A. Moretta, L. Moretta, T. R. Mosmann, S. Müller, W. Müller, C. Münz, G. Multhoff, L. E. Munoz, K. M. Murphy, T. Nakayama, M. Nasi, C. Neudörfl, J. Nolan, S. Nourshargh, J.-E. O'Connor, W. Ouyang, A. Oxenius, R. Palankar, I. Panse, P. Peterson, C. Peth, J. Petriz, D. Philips, W. Pickl, S. Piconese, M. Pinti, A. G. Pockley, M. J. Podolska, C. Pucillo, S. A. Quataert, T. R. D. J. Radstake, B. Rajwa, J. A. Rebhahn, D. Recktenwald, E. B. Remmerswaal, K. Rezvani, L. G. Rico, J. P. Robinson, C. Romagnani, A. Rubartelli, B. Ruckert, J. Ruland, S. Sakaguchi, F. S. de Oyanguren, Y. Samstag, S. Sanderson, B. Sawitzki, A. Scheffold, M. Schiemann, F. Schildberg, E. Schimisky, S. A. Schmid, S. Schmitt, K. Schober, T. Schüler, A. R. Schulz, T. Schumacher, C. Scotta, T. V. Shankey, A. Shemer, A.-K. Simon, J. Spidlen, A. M. Stall, R. Stark, C. Stehle, M. Stein, T. Steinmetz, H. Stockinger, Y. Takahama, A. Tarnok, Z. Tian, G. Toldi, J. Tornack, E. Traggiai, J. Trotter, H. Ulrich, M. van der Braber, R. A. W. van Lier, M. Veldhoen, S. Vento-Asturias, P. Vieira, D. Voehringer, H.-D. Volk, K. von Volkmann, A. Waisman, R. Walker, M. D. Ward, K. Warnatz, S. Warth, J. V. Watson, C. Watzl, L. Wegener, A. Wiedemann, J. Wienands, G. Willimsky, J. Wing, P. Wurst, L. Yu, A. Yue, Q. Zhang, Y. Zhao, S. Ziegler, and J. Zimmermann, "Guidelines for the use of flow cytometry and cell sorting in immunological studies," *European Journal of Immunology*, vol. 47, pp. 1584–1797, Oct. 2017.

[13] M. Monici, "Cell and tissue autofluorescence research and diagnostic applications," in *Biotechnology Annual Review*, pp. 227–256, Elsevier, 2005.

[14] N. S. Barteneva and I. A. Vorobjev, eds., *Imaging Flow Cytometry*. Springer New York, 2016.

[15] M. J. Sanderson, I. Smith, I. Parker, and M. D. Bootman, "Fluorescence microscopy," *Cold Spring Harbor Protocols*, vol. 2014, pp. pdb.top071795–

pdb.top071795, Oct. 2014.

[16] E. Lane and C. M. Alexander, "Use of keratin antibodies in tumor diagnosis.," *Seminars in cancer biology*, vol. 1 3, pp. 165–79, 1990.

[17] J. Kapuscinski, "DAPI: a DNA-specific fluorescent probe," *Biotechnic & Histochemistry*, vol. 70, pp. 220–233, Jan. 1995.

[18] V. Barak, H. Goike, K. W. Panaretakis, and R. Einarsson, "Clinical utility of cytokeratins as tumor markers," *Clinical Biochemistry*, vol. 37, pp. 529–540, July 2004.

[19] D. Maetzel, S. Denzel, B. Mack, M. Canis, P. Went, M. Benk, C. Kieu, P. Papior, P. A. Baeuerle, M. Munz, and O. Gires, "Nuclear signalling by tumour-associated antigen EpCAM," *Nature Cell Biology*, vol. 11, pp. 162–171, Jan. 2009.

[20] A. Armstrong and S. L. Eck, "EpCAM: A new therapeutic target for an old cancer antigen," *Cancer Biology & Therapy*, vol. 2, pp. 320–325, July 2003.

[21] A. Rheinländer, B. Schraven, and U. Bommhardt, "CD45 in human physiology and clinical medicine," *Immunology Letters*, vol. 196, pp. 22–32, Apr. 2018.

[22] J. Yoon, A. Terada, and H. Kita, "CD66b regulates adhesion and activation of human eosinophils," *The Journal of Immunology*, vol. 179, pp. 8454–8462, Dec. 2007.

[23] T. Schmidt, A. Brodesser, N. Schnitzler, T. Grüger, K. Brandenburg, J. Zinserling, and J. Zündorf, "CD66b overexpression and loss of c5a receptors as surface markers for staphylococcus aureus-induced neutrophil dysfunction," *PLOS ONE*, vol. 10, July 2015.

[24] A. N. Mayeno, K. J. Hamann, and G. J. Gleich, "Granule-associated flavin adenine dinucleotide (FAD) is responsible for eosinophil autofluorescence," *Journal of Leukocyte Biology*, vol. 51, pp. 172–175, Feb. 1992.

[25] K. Miura and J. Schindelin, *ij_Textbook1: Updates From 2015 (Version v2.1.3)*. Zenodo, 2016.

[26] J. Ghaye, M. A. Kamat, L. Corbino-Giunta, P. Silacci, G. Vergères, G. D. Micheli, and S. Carrara, "Image thresholding techniques for localization of sub-resolution fluorescent biomarkers," *Cytometry Part A*, vol. 83, pp. 1001–1016, Sept. 2013.

[27] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan. 1979.

[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. New York, NY: Pearson, 4 ed., 2018.

[29] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, pp. 113–125, July 1994.

[30] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, pp. 176–201, Apr. 1993.

[31] M. Linkert, C. T. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. MacDonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, and J. R. Swedlow, "Metadata matters: access to image data in the real world," *Journal of Cell Biology*, vol. 189, pp. 777–782, May 2010.

[32] S. Ota, I. Sato, and R. Horisaki, "Implementing machine learning methods for imaging flow cytometry," *Microscopy*, vol. 69, pp. 61–68, Mar. 2020.

[33] MATLAB, "regionprops," accessed 2020-12-04, in MATLAB version R2020b Documentation, url: https://se.mathworks.com/help/images/ref/regionprops.html.

[34] vega, "Voyager 2," accessed 2020-12-04, on GitHub, url: https://github.com/vega/voyager.

[35] H. Hennig, P. Rees, T. Blasi, L. Kamentsky, J. Hung, D. Dao, A. E. Carpenter, and A. Filby, "An open-source solution for advanced imaging flow cytometry data analysis using machine learning," *Methods*, vol. 112, pp. 201–210, Jan. 2017.