Image analysis for gene expression based phenotype characterization in yeast cells

Tleis, M.

Author: Tleis, Mohamed
Title: Image analysis for gene expression based phenotype characterization in yeast cells
Issue Date: 2016-07-06

2

Yeast Analysis Platform

> *This chapter describes the yeast analysis framework we have developed to support research in yeast biology. We start by presenting the workflow of the system and elaborate on its various modules including segmentation, measurement, analysis and visualization, and finally the Graphical User Interface (GUI) that connects all these modules. A validation small scale experiment is performed as well.*

This chapter is based on the following publications:

- Mohamed Tleis, Ginny Anemaet, Paul van Heusden, and Fons J. Verbeek. Image analysis platform for yeast biologists. In The 2nd International Conference on Advances in Biomedical Engineering 2013 (ICABME'13), pages 57–60, Tripoli, Lebanon, September 2013.
- Mohamed Tleis and Fons J. Verbeek. Yeast-Cell features extraction plugin. In ImageJ Conference 2012, Mondorf, Luxembourg, October 2012.

2.1 Background

THE goal of this study is to address which components and processes can form an objective and comprehensive image analysis pipeline to analyze image-based gene expression in single cells. The developed objective image analysis platform should be applicable for live cell imaging, thus without additional staining and should perform in an automatic way the entire image analysis pipeline:

- Fluorescence (confocal) microscopy images.
- Detection of individual cells on the images (segmentation).
- Quantification of the fluorescent signal and other characteristics of each individual cell (measurement).
- Statistical analysis of the measurements.
- Visualization of the measurements in several graphical ways.

To quantify expression levels of GFP-tagged reporter genes in fluorescent microscope images, several software tools have been developed. The first step is to detect the individual cells in the image through segmentation. Detection of cells can be performed both in the bright-field channel as well as in the fluorescence channel(s). Until recent, a software tool has been used for screening the yeast GFP-fusion library to investigate global cellular protein reorganization on exposure to the alkylating agent methyl-methane-sulfonate [Maz13]. In that image analysis software, developed in Matlab, cellular boundaries are detected after staining the cell-wall with Alexa 647 conjugated Concanavalin A. A disadvantage of such an approach is that additional staining is required. Another recent research implements a pipeline including high-throughput microscopy, automated image analysis, and pattern classification through machine learning [Cho15]. The approach followed in that research also requires staining. It uses yeast synthetic genetic array (SGA) technology to introduce a cytosolic red fluorescent protein (RFP) to mark cell boundaries. Moreover, a number of software tools that processes bright-field images have been described, such as Pombex [Pen13], CellStat [Kva08] and CellSerpent [Bre11]. Pombex segments *Schizosaccharomyces pombe* cells in bright-field images. For *S. cerevisiae*, however, Pombex is not optimal as they have different shape characteristics. CellStat is able to segment bright-field images of *S. cerevisiae*, but it has a constraint on the cells to be detected, as they must not be encapsulated by other cells [Kva08]. CellSerpent utilizes an active contour segmentation algorithm for cell detection, where only few features are measured [Bre11]. Moreover, both CellStat and CellSerpent requires the Matlab software to be installed. In another study, images obtained by confocal microscopy are analyzed to investigate the localization of the plasma membrane protein *Mrh1p-GFP* [Bir11]. Image analysis is carried out using modules derived from the Acapella software that is supplied with the Opera microscope that the software is developed for, and hence the modules are not freely available.

As expression levels and subcellular localization of tagged genes are highly variable, we will use bright-field for the detection of cells. In addition, the information from the fluorescent datasets is given. Nevertheless, the software offers segmentation methods that works well with fluorescence images when needed. In the next section, we describe the workflow of our developed platform. The platform is named *YeastAnalysis*.

2.2 YeastAnalysis Workflow

Image analysis was probed to accomplish further progress to complement flow cytometric analysis for the *S. cerevisiae* yeast cells. This should be part of a comprehensive image analysis platform. Such platform must have the following components: (1) Segmentation, (2) Measurement, (3) Data analysis and visualization and (4) a graphical user interface (*GUI*).

The proposed workflow is depicted in Fig. 2.1. The three major components offered by the system are shown as separate blocks connected through the *GUI*. In the next sections we elaborate on each of these modules including segmentation, measurement, data analysis and visualization and the *GUI*. Subsequently, a small scale validation experiment is presented in Section 2.7. Section 2.8 discusses the hardware and software tools we used during the development of the platform. We complete this chapter with a conclusion in Section 2.9.

2.3 Segmentation Module

For different image modalities, different segmentation methods give different results. In our research we identify four different type of images. In Fig. 2.1 we label these images as Fluorescent Type A, Fluorescent Type B, Bright-field Type C and Bright-field Type D. Four different segmentation processes are developed for each image type. As a general approach we always follow the same segmentation path used with Type D; i.e. applying *Hough Transform* and *Minimal Path* algorithms as the core methods. However, when results are not satisfactory, one can always choose a different method according to the image modality. Type A fluorescent images are those whose cells show evenly distributed fluorescent signal throughout the cell. Example of this type are images showing the expression of *BMH* gene tagged with *GFP* as a reporter gene. Type B fluorescent images are those whose cells show strong signal throughout the cell or nucleus and possesses speckle noise. For example, images showing the nucleus stained with *DAPI* signal, and the cytoplasm of the cell shows speckle noise throughout the cytoplasm possibly corresponding to mitochondria. Type C images uses brightfield channel where the cell structures are well separated and their contours are well highlighted. Example of this type, is when the fluorescent channel is weak and its segmentation is not possible with standard methods; then we use this brightfield channel to detect the cell objects. Type D images represent the most sophisticated case, where the fluorescent signal is weak, and the brightfield channel has cells clumped within other cells, or attached to budding cells, leading to disconnected cell contours.

In this section, we discuss two types of segmentation algorithms used by the system. The first is filter-based and the second uses *Hough Transform* and *Minimal Path* algorithms.

2.3.1 Filter-Based Methods

In these type of segmentation methods, we use one of three different filters as its core processing followed by triangle auto-threshold, fill holes and morphological watershed algorithms. The three different filters are the Sigma, Despeckle and Sobel. These filters will be discussed first. Subsequently we discuss the triangle auto-threshold, fill holes and morphological watersheds.

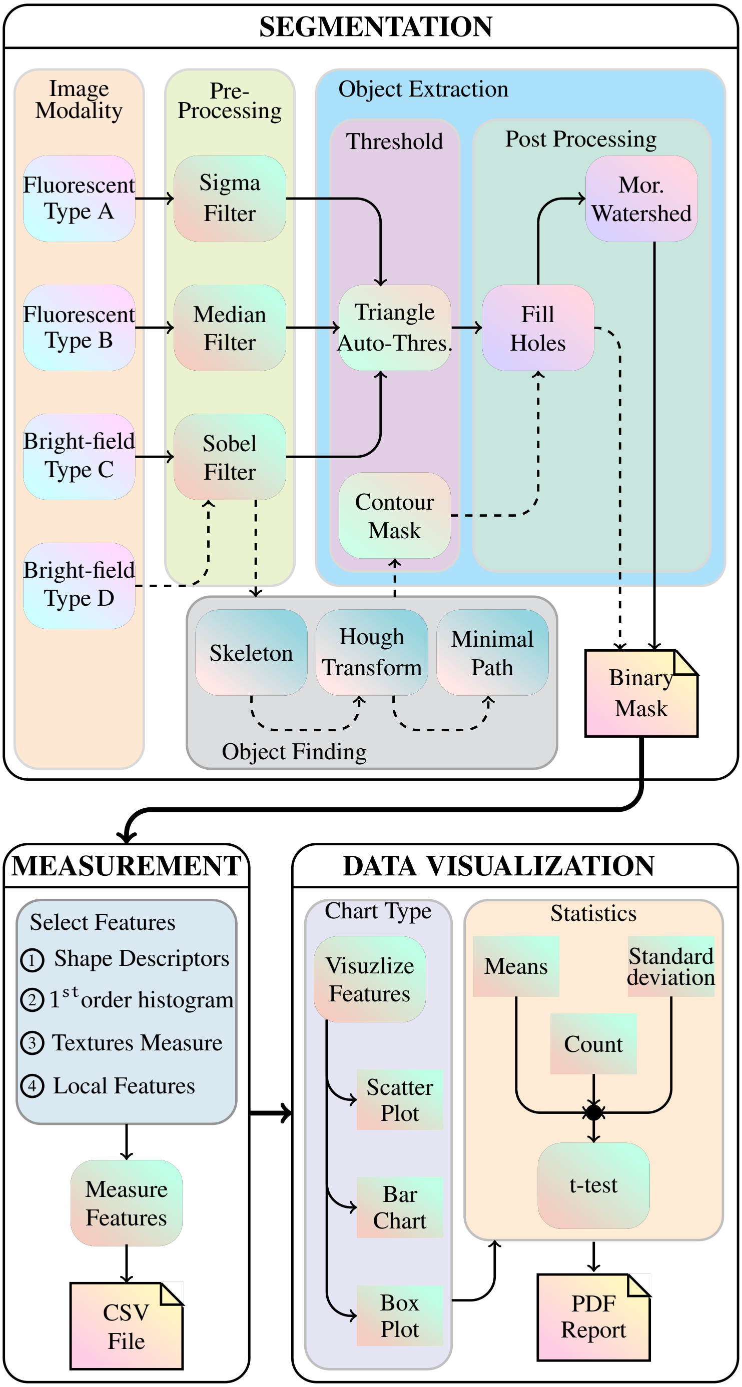


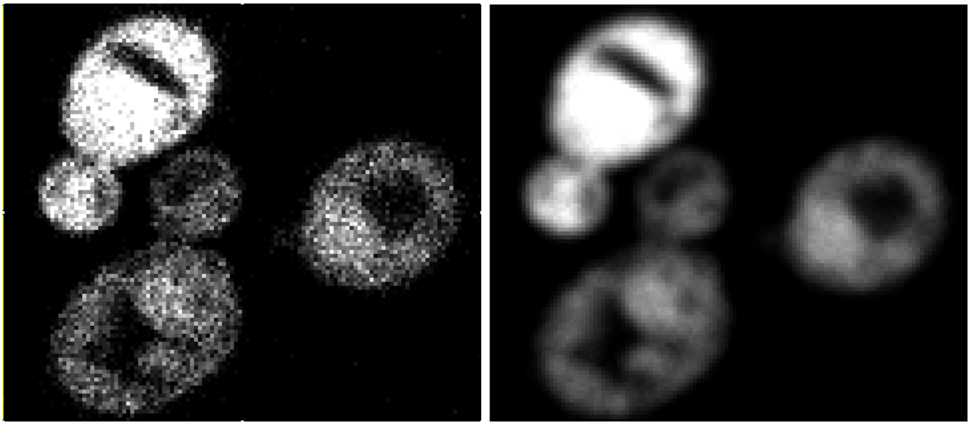
Figure 2.1: *Workflow of the YeastAnalysis Platform.*

Sigma Filter

We use Sigma filter [Lee83, Sch07] in one of the simple segmentation methods on fluorescent images. This filter is motivated by the sigma probability of the Gaussian distribution, and it smooths the image noise by averaging only those neighbourhood pixels that have the intensities within a fixed sigma range of the center pixel. Smoothing and blurring the images through Sigma filter makes it possible to acquire binary masks that better represent the shape of the objects. It was selected for a number of advantages including:

- Noise near edge areas will be smoothed without blurring the edge because only pixels on one side of the edge are included in the average;
- Preservation of subtle details and linear features;
- Not sensitive to shape distortion;
- Retention of step edges and sharpening of ramp edges;
- Removal of high-contrast spot noise;
- Computationally efficient.

Figure 2.2 illustrate a sample application of sigma filter on *S. cerevisiae* cells.



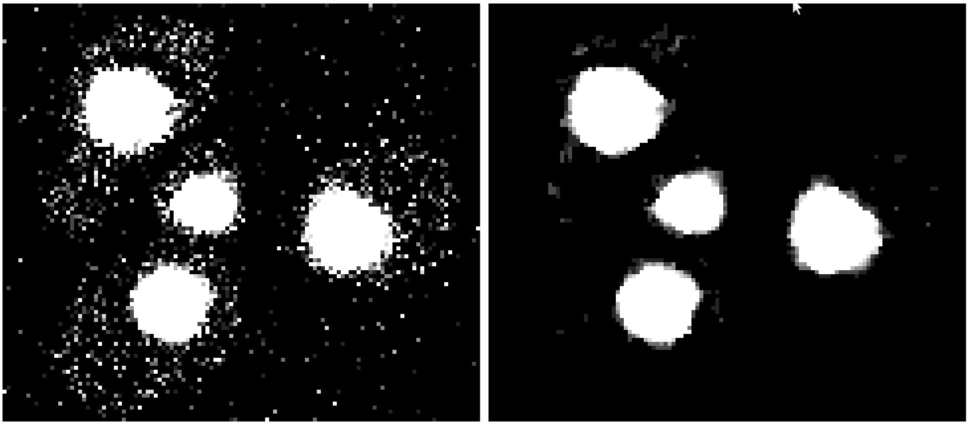
(a) Fluorescence Image

(b) Applying Sigma Filter

Figure 2.2: *Applying Sigma Filter on a Fluorescence Image.*

Median Filter

In some fluorescent images, there exists speckle noise, also known as salt-and-pepper kind of noise. Since Median filters are well known as a good approach to remove such kind of noise [Gon08], we apply a Despeckle algorithm [Ras16], which is a median filter that replaces each pixel with the median value in its 3 x 3 neighbourhood. Figure 2.3 shows a threshold image of yeast cell nuclei before and after the application of the Despeckle algorithm.



(a) impulsive noise

(b) Application of a Median Filter

Figure 2.3: *Median Filter.*

Sobel Filter

Sobel filter is a discrete differentiation operator, computing an approximation first estimate of the gradient of the image intensity function. We use the Sobel filter to highlight and detect the edges of the cells. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. This vector has the important geometrical property that it points to the direction of the greatest rate of change at a certain location (x, y) in the image. The Sobel operator is based on convolving the image with a 3x3 filter masks. These masks are separable and integer valued in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations [Gon08, Ras16].

Triangle auto threshold

In the filter-based segmentation methods, the Triangle auto-threshold [Zac77, Ras16] is used to obtain a binary image as shown in Fig. 2.5(a). The threshold level in the Triangle auto threshold method is determined on the basis of the histogram of pixel intensities as illustrated in Fig. 2.4.

We evaluate the distance from the histogram at every level to the hypotenuse of the triangle having the histogram height and dynamic range as sides. The histogram level having the maximum distance corresponds to the final threshold used by this method. In Fig. 2.4, point T corresponds to such threshold.

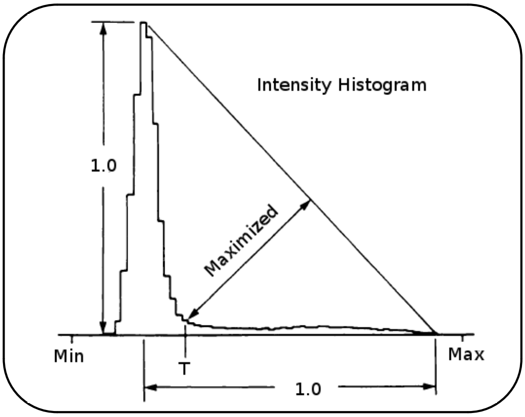


Figure 2.4: *Triangle auto threshold.*

The microscope we used throughout our research produce images whose histogram has a weak bimodal distribution and a high kurtosis as it has a distinct peak near the mean, which belongs to the low extreme of the histogram. Triangle's method has proved its superiority for these kind of images as it assumes a peak near one of the extremes and searches the threshold value toward the other end [Car12]. Additionally, Triangle's method is applied once on the histogram and it is computationally in-expensive unlike some other threshold methods such as the Otsu and Isodata.

The final result of the threshold operation is a binary image. Sometimes some inner parts of the objects are not recognized as foreground pixels. Hence, we apply *Fill Holes* algorithm.

Fill Holes

After obtaining a binary image through threshold, the *Fill Holes* algorithm simply walkthrough the pixels starting from the boundaries of the image and filling all the pixels with a background label and stops when facing a foreground pixel. Labelling all the background pixels leaves all the closed contours and their inner regions to be considered as foreground pixels. This operation produces a binary mask of the cells. To separate the cells that are connected together, we apply the morphological watersheds discussed hereafter.

Morphological watersheds

As Fig. 2.5(a) shows, the obtained binary image might have cells clumped together. For separation of such clumped cells, *morphological watersheds* [Gon08, Ras16] is used to obtain the final binary mask of the cells. Figure 2.5(b) shows the same cells from Fig. 2.5(a) processed with *watersheds*. The concept of *watersheds* is based on visualizing an image in three dimensions: two spatial coordinates versus intensity. In such a "topographic" interpretation, we consider three types of points: (a) points belonging to a regional minimum; (b) points at which a drop of water, if placed at the location of any of those points, would fall with certainty to a single minimum; and (c) points at which water would be equally likely to fall to more than one such minimum. For a particular regional minimum, the set of points satisfying condition (b) is called the *catchment basin* or *watershed* of that minimum. The points satisfying condition (c) form crest lines on the topographic surface and are termed *divide lines* or *watershed lines* [Gon08]. These *watershed lines* are located by the algorithm to separate the clumped *S. cerevisiae* cells.

2.3.2 Hough Transform and Minimal Path

The second type of segmentation methods is developed to be used as a general segmentation algorithm on bright-field images and it uses our novel *Hough Transform* and *Minimal Path* algorithms in its core. The idea of the new method is to use *Hough Transform* to locate the geometrical circles that contains part of the cell contours in a skeleton of the gradient image where the foreground shapes are reduced to a skeleton of one pixel width. Subsequently, a polar transformation is applied to resample the pixels surrounding the cells, and apply *Minimal*

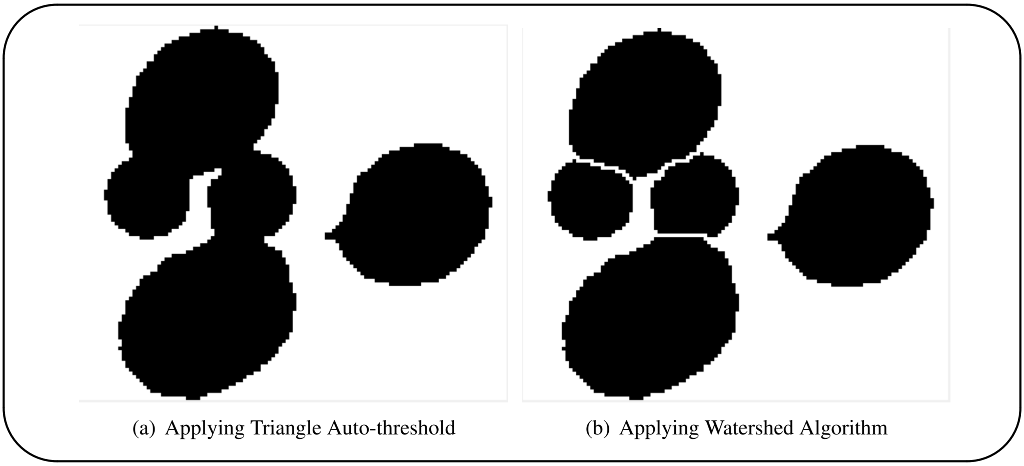

(a) Applying Triangle Auto-threshold
(b) Applying Watershed Algorithm

Figure 2.5: *Threshold Image and Separating Cells.*

Path algorithm to find the path (full contour) of every cell. Chapter 3 explains more about this method. In addition, Chapter 3 describes an extension for the *Hough Transform* and *Minimal Path* algorithm by adding a contour expansion process. This expansion offers more optimized contours in some image modalities. The extracted contours are then used to obtain a binary mask of every yeast cell, enabling its measurement in the overlaid fluorescence channel.

After the segmentation process, the extracted contours or the binary masks are used to measure the individual *S. cerevisiae* yeast cells. The measurement is performed within the Measurement module in the image analysis pipeline. The Measurement module is discussed hereafter.

2.4 Measurement Module

This part of the workflow measures and describes the *S. cerevisiae* yeast cells for various features and textures. Using the labelled objects from the binary masks generated during the segmentation process, measurement of individual cells is made possible. This system provides an option to choose from a list of features to be measured; moreover, it provides an option to exclude outliers based on the values of the circularity feature and size of the measured cells.

In this platform, basic feature extraction techniques in image analysis are considered, while more sophisticated techniques are studied later in Chapter 4. We categorize the basic techniques used in this platform into two classes. The first is based on first-order statistics and the second is based on texture measurement. In the following sub-section we start discussing the first-order statistic features; subsequently, we discuss the basic texture measurements.

2.4.1 First order statistics based features

By first-order statistic features, we mean all those features computed based on single pixel values including first-order histogram based features. This is in addition to basic shape and intensity descriptors.

Assuming that our microscope image is a function $f(x, y)$ of two space variables x and y, $x = 0, 1, \ldots .N - 1$ and $y = 0, 1...M - 1$. The function $f(x, y)$ can take discrete values $i = 0, 1...L - 1$, where L is the total number of intensity levels in the image. The intensity-level histogram is a function showing the number of pixels for each intensity level in the whole image. This function is depicted in Eq. 2.1.

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x, y), i), \tag{2.1}$$

where $\delta(j, i)$ is the Kronecker delta function, depicted in Eq. 2.2. The Kronecker delta function simply increments the intensity level histogram by the value of one at every pixel whose intensity value $j = f(x, y)$ is equal to that histogram intensity level i.

$$\delta(j, i) = \begin{cases} 1, j = i \\ 0, j \neq i \end{cases} \tag{2.2}$$

The histogram of intensity levels is obviously a concise and simple summary of the statistical information contained in the image. Calculation of the grey-level histogram involves all single pixels. The histogram contains the first-order statistical information about the image. The histogram can also be computed for a sub-image, i.e. the region of interest (RoI) of cell objects. Different useful image features are worked out from the histogram to quantitatively describe the first-order statistical properties of the objects. In this research, we considered many basic shape and intensity descriptors based on the first order statistical information in the images. A list of those features is depicted in Table 2.1.

2.4.2 Texture Measurement

An image or object texture is an important metric as it gives us information about the spatial arrangement of intensities in that image or selected region of an image, i.e. the object. A set of basic texture features [Gon08] is available to be measured in the measurement module. A list of these texture features is depicted in Table 2.2.

All these measurements of features and simple feature textures are saved automatically into a CSV (comma-separated values) file that is used at the following step to generate a report,

Table 2.1: Features based on first order statistics.

Features	Description
Size	The size of a yeast cell (object) is simply the number of pixels occupied by that cell in the image, and for SI unit, it is multiplied by the area of one pixel in square µ-meters (width x height of pixel).
Total Intensity	The total intensity of a yeast cell is the total gray-level values of the pixels occupied by this cell, which ranges between 0 and 255 for the gray images used.
Intensity Standard Deviation	The standard deviation from the mean of the intensity values in each cell.
Perimeter	The perimeter representation method used to estimate the perimeter of the cell is that of Vossepoel and Smeulders [Vos82]. This representation is based on chaincodes: $L_{vs} = (0.980).N_e + (1.406).N_o - (0.091).N_c \qquad (2.3)$ where N_e, N_o and N_c represents the number of even codes, odd codes and corners in the chaincode of cell boundaries [Gon08].
Density	The density of a particle object is its area multiplied by the average mean of gray-level values: $d = A * \mu \qquad (2.4)$ where *d* represents density, *A* represents area and µ represents mean of intensity of a measured cell.
Circularity	The circularity of detected shapes [Ras16]. $Circularity = \frac{4\pi x Size}{perimeter^2}. \qquad (2.5)$
Vacuole Size	If the fluorescent protein is expressed by genes known to have expression in the cytoplasm and nucleus without its vacuoles, the size of the central vacuole can be estimated. This is computed by using a vacuole filter algorithm that looks in the fluorescent images for a region, inside the RoI (region of Interest) representing every cell, that forms the largest connected region with the lowest intensity values.
Membrane Features	Different features can be measured in the region close to the cell borders where membrane proteins are expressed. Such features include size, total Intensity and Intensity standard deviation.
Nucleus Features	For those images that contain a nuclear stain such as DAPI, the nucleus can be measured for all the features an individual cell could be measured for.

Table 2.2: Basic texture measurement.

Features	Description
Variance	The variance (μ_2 or σ^2) is a measure of intensity contrast and can be computed from the second statistical moment. $$\mu_2(z) = \sum_{i=0}^{L-1} (z_i - m)^2.P(z_i) \qquad (2.6)$$ where z_i is the intensity value of the histogram at location i, m is the mean intensity value, L is the total number of intensity levels (histogram range), and $P(z_i)$ is the corresponding histogram with i between 0 and $L - 1$ [Gon08]).
Smoothness	The variance (σ^2) is used to establish the descriptor of relative smoothness (S): $$S(z) = 1 - \frac{1}{1 + \sigma^2(z)} \qquad (2.7)$$ This measure is zero for areas of constant intensities where the variance is zero there, and it approaches 1 for large values of the variance [Gon08].
Skewness	The skewness (μ_3) of the intensity histogram which is the third statistical moment. $$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3.P(z_i) \qquad (2.8)$$ A negative skewness means that most of the pixel values are high and thus concentrated at the right side of the histogram. A positive skewness means that most of the pixel values are low and thus concentrated at the left side of the histogram [Gon08].
Uniformity	The uniformity (U) has a maximum value for a cell image in which all intensity levels are equal [Gon08]. $$U(z) = \sum_{i=0}^{L-1} P^2(z_i) \qquad (2.9)$$
Entropy	The Entropy (e), which is a measure of variability, is zero for constant images [Gon08]. $$e(z) = -\sum_{i=0}^{L-1} P(z_i).\log_2 P(z_i) \qquad (2.10)$$

or to be inspected manually. More advanced sophisticated feature extraction techniques are considered in our research to improve the recognition of objects after the segmentation process and to discriminate yeast cells into different categories with the purpose of identifying subtle patterns. These sophisticated features are explained in Chapter 4.

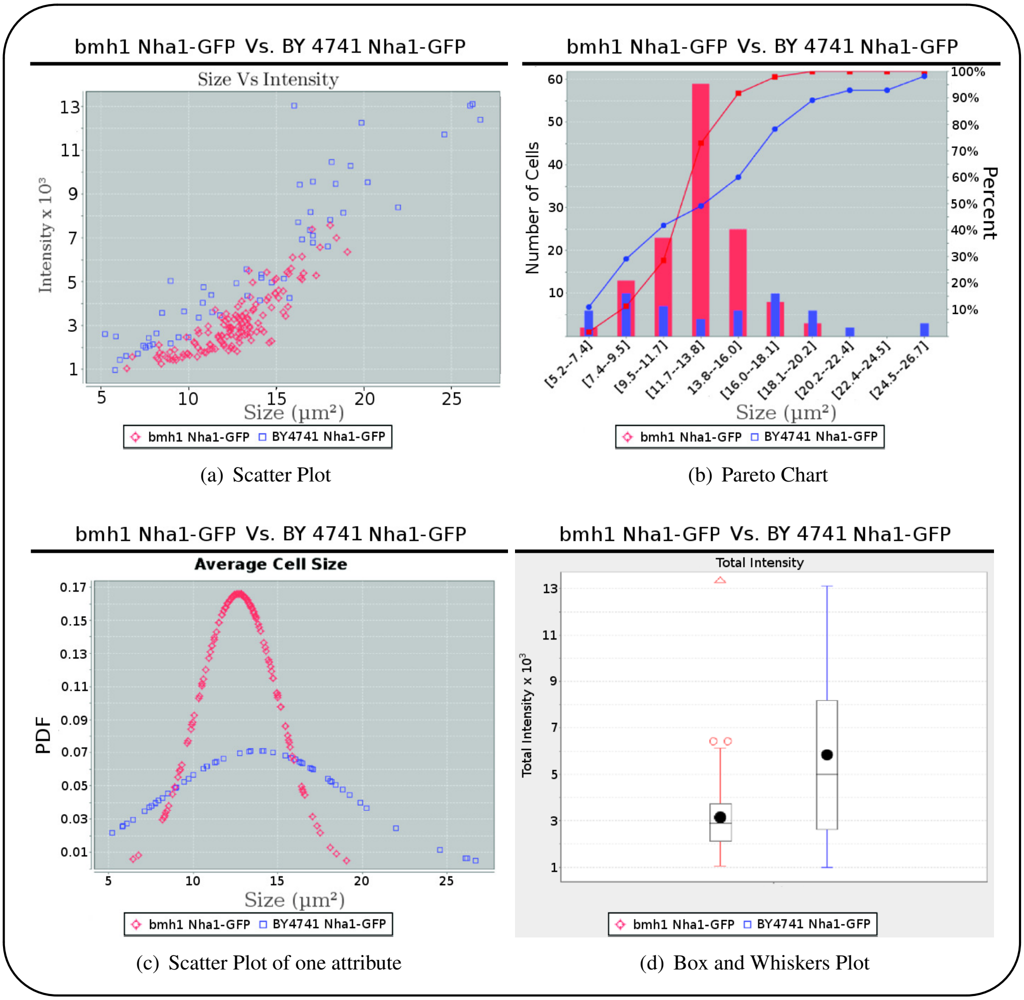
2.5 Data Analysis and Visualization Module

After measurement, usually molecular biologists would like to compare two classes of cells to study any significant changes in cell size or fluorescence intensity in cells having a certain gene mutation or cultured in a different medium (ex. 2M NaCl, 50mM KCl, etc...). The changes in features of cells in different conditions are compared usually to wild-type cells. To assist in achieving this goal of comparison, our system generates automatically a report in pdf format including basic statistics about the experiment and graph charts to visualize the results.

The statistical information created into the report includes counts of the cells belonging to different cultures, the mean value of their surface area, and fluorescence intensity along with its standard deviations. The unpaired student t-test is performed to report the t-value and p-value to assist in recognizing the significant of the difference between two different cell samples.

The module in the yeast analysis pipeline performs statistical descriptive data analysis. More advanced analysis of the measurement data is considered in Chapter 4. Such advanced analysis makes use of machine learning techniques. This machine learning approach is used to improve the segmentation process. It also makes use of a similar approach to discriminate various cell conditions aiming at identifying subtle patterns.

The process of data analysis would not provide significant details for users without some visualization tools. This platform offers some data visualization techniques to assist biologists in extracting meaningful information from their experiments. A very frequently used graph chart in yeast cell biology are those scatter plots that visualize the different values of cell surface area on one axis against fluorescence intensity on the other, with a different label for each cell culture. Figure 2.6(a) depicts a sample of such plot. A scatter plot is typically used to compare between two observations represented by two variables to study their correlation, and determine if the two variables exhibit the same or opposite direction. Other visualization options available are charts visualizing one attribute of the cells in one chart, as a Pareto chart (cf. Fig. 2.6(b)), a scatter plot visualizing the Gaussian density distribution of the data for the two cell cultures (cf. Fig. 2.6(c)), or a box and whiskers plot to facilitate the analysis of the range of data (cf. Fig. 2.6(d)).



(a) Scatter Plot

(b) Pareto Chart

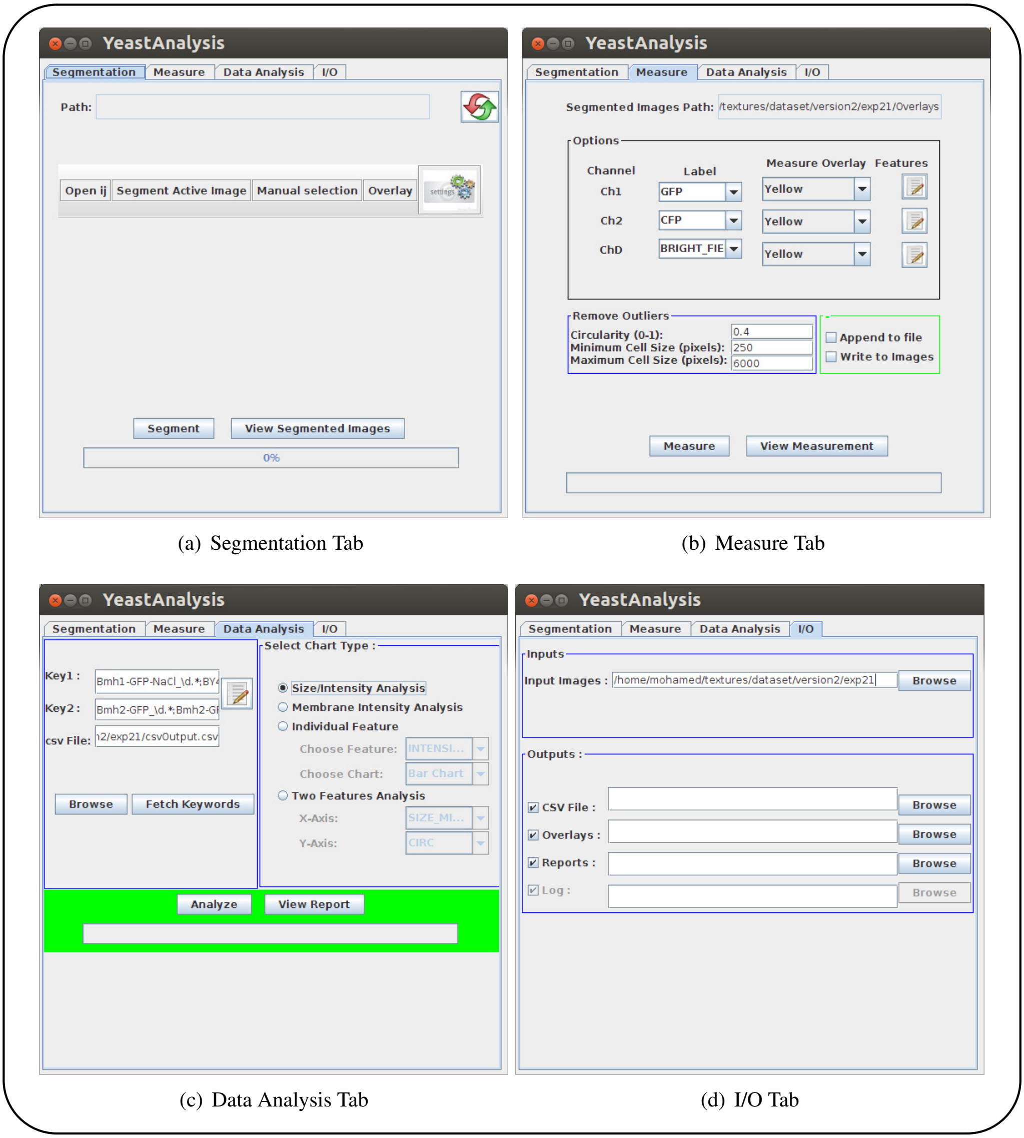
(c) Scatter Plot of one attribute

(d) Box and Whiskers Plot

Figure 2.6: *Graph Charts to visualize measurement.*

2.6 System GUI

The graphical user interface is a necessary tool that connects the workflow components together. In its basic structure, the *GUI* is composed of the different modules mentioned in the workflow in Fig. 2.1. These modules are provided as separate tabs in a user friendly interface. Figure 2.7 depicts the interfaces of the *GUI*. Figure 2.7(a) shows the interface where a user can set the segmentation method and parameters to perform the segmentation process. Figure 2.7(b) depicts the interface in the measure tab, where users can specify the channels and features to be processed. Figure 2.7(c) shows the data analysis part, where one can set multiple pairs of keywords corresponding to different cell strains or cultures. The last tab in the *GUI* is an interface to set the directory of the input images as well as the output directories of the segmentation, measurement and analysis results. Besides being implemented as a stand-alone desktop application, this *GUI* was also implemented as a plugin for the imageJ software [Tle12].



Path:

0%

(a) Segmentation Tab

Segmented Images Path: /textures/dataset/version2/exp21/Overlays

Options

Channel	Label	Measure Overlay	Features
Ch1	GFP	Yellow	
Ch2	CFP	Yellow	
ChD	BRIGHT_FIE	Yellow	

Remove Outliers

Circularity (0-1): 0.4
Minimum Cell Size (pixels): 250
Maximum Cell Size (pixels): 6000

□ Append to file
□ Write to Images

(b) Measure Tab

Key1 : Bmh1-GFP-NaCl_\d.*;BY4

Key2 : Bmh2-GFP_\d.*;Bmh2-GF

csv File: ı2/exp21/csvOutput.csv

Select Chart Type :

● Size/Intensity Analysis
○ Membrane Intensity Analysis
○ Individual Feature
Choose Feature: INTENSI...
Choose Chart: Bar Chart
○ Two Features Analysis
X-Axis: SIZE_MI...
Y-Axis: CIRC

(c) Data Analysis Tab

Inputs

Input Images : /home/mohamed/textures/dataset/version2/exp21 Browse

Outputs :

☑ CSV File : Browse
☑ Overlays : Browse
☑ Reports : Browse
☑ Log : Browse

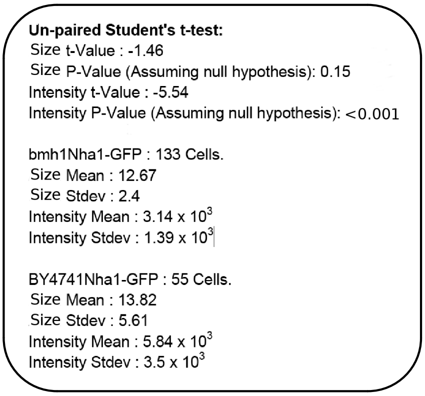
(d) I/O Tab

Figure 2.7: *GUI of the Yeast Analysis platform.*

2.7 Workflow validation

To validate *YeastAnalysis* platform, a small scale experiment is performed to compare two cell cultures. The first sample belonging to a strain of cells having a mutation in the *BMH1* gene and expressing *Nha1* protein attached to a green fluorescent protein (*GFP*). This cell strain is labelled *Δbmh1 Nha1-GFP*. The second sample is from a wildtype cell strain expressing *Nha1* protein attached to *GFP*. This strain is labelled *BY4741 Nha1-GFP*. A set of 18 images are acquired with a Zeiss LSM5 Exciter confocal microscope. From the segmentation, 188 cells are detected using our novel segmentation method on the bright-field image channels; i.e. *Hough Transform* followed by Circular Shortest Path (*HCSP*). More details about this method are explained in Chapter 3. Ensuing segmentation, the cells are measured for various features and all the measurements are written to a CSV file. The measurements are then analyzed automatically to generate a report including graph charts and statistics. Such charts are shown in Fig. 2.6. Example of the statistical information automatically generated into the report is shown in Fig. 2.8.

Having a look at the chart and the statistics, and specifically the p-values of the cell size and intensity, reveals meaningful information. It is clearly visible that mutant cells belonging to the *Δbmh1 Nha1-GFP* strain are smaller and have less *GFP* fluorescence than the wildtype cells in *BY4741 Nha1-GFP* strain. The results obtained using the image analysis platform we developed in our research are in good correspondence with the flow cytometry test. Figure 2.9 reveals these results obtained by flow cytometry.



Un-paired Student's t-test:
Size t-Value : -1.46
Size P-Value (Assuming null hypothesis): 0.15
Intensity t-Value : -5.54
Intensity P-Value (Assuming null hypothesis): <0.001

bmh1Nha1-GFP : 133 Cells.
Size Mean : 12.67
Size Stdev : 2.4
Intensity Mean : 3.14 x 10^3
Intensity Stdev : 1.39 x 10^3

BY4741Nha1-GFP : 55 Cells.
Size Mean : 13.82
Size Stdev : 5.61
Intensity Mean : 5.84 x 10^3
Intensity Stdev : 3.5 x 10^3

Figure 2.8: *Sample report from the experiment analysis.*

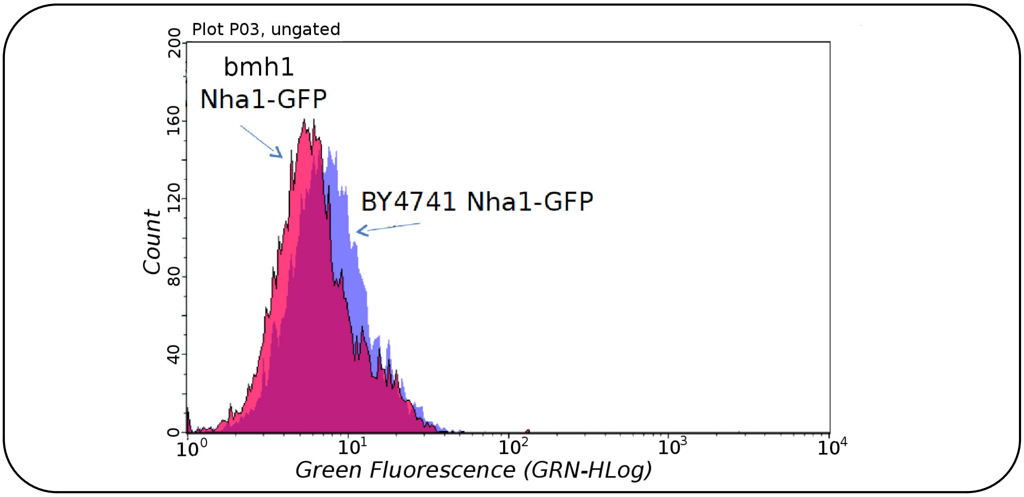


Figure 2.9: *Results of the Flow Cytometry test.*

2.8 Development of *YeastAnalysis*

The hardware used in the development of *YeastAnalysis* is a desktop computer equipped with an Intel® Core™ i7-2600 CPU @ 3.40GHz × 8 processor, and 16 GB of memory installed with the Ubuntu 12.04 LTS operating system. The software is written in Java using JDK-1.7 including a variety of open-source packages. An ImageJ library package [Ras16] is utilized to access some existing functions such as the Watershed Algorithm, Fill Holes Algorithm and Overlays feature. ImageJ plugins are also imported, such as the Sigma-Filter-Plus which is an implementation of the sigma filter [Sch07]. The output of measurement was integrated with spreadsheet applications through Java Excel API [jex16]. The t-test and statistical analysis of these measurements are performed with the support of the Apache commons Mathematics library [apa16]. Visualization of statistical results was made possible with the JfreeChart API [jfr16]. Creation of the Pdf reports was done with iText programmable pdf software [ite16]. Moreover, for drawing of the ground-truth images we used the TDR software package [Ver04]. *YeastAnalysis* and a manual are available for download from [git16].

2.9 Conclusion

In this chapter, we addressed the components and processes that builds up an automatic image analysis pipeline for the objective analysis of gene expression in single cells, i.e. yeast cells. We designed a workflow for this pipeline. This workflow is composed of a segmentation module, a measurement module, a data analysis and visualization module and a *GUI* connecting the aforementioned modules.

The segmentation module has various segmentation and image processing algorithms adopted in this system including the sigma filter based method, median filter based method, Sobel filter based method and *Hough Transform* and *Minimal Path* algorithms. Including as well the post-processing methods such as Triangle, fill holes and watershed. After the segmentation phase, there is the measurement phase. The measurement module has various features that could be selected and measured. These features are categorized into first-order statistical features and basic texture measurements based on the histogram intensities. After the measurement phase, comes the data analysis and visualization phase. In this part, the statistical descriptive data analysis that is performed include the computation of basic statistical metrics and Student's t-test for statistical analysis. The *GUI* of the developed image analysis platform connects the various modules as separate tabs. There is also a room to add new components. Finally, the platform is validated in a small scale experiment; the analysis of a study experiment revealed decreased expression of the *Nha1* protein in a yeast strain having a mutation in the *BMH1* gene compared to the wildtype strain.

The results demonstrate that this platform can potentially contribute to the improvement of the objective analysis and diagnosis of gene expression studies in yeast. It reveals an advantageous comprehensive image analysis platform that can be used in the laboratory assisting in analyzing experiments.