Automated Detection of Mating Phenotype in *Saccharomyces Cerevisiae*

By: Brennan Fife (bfife@wisc.edu), James Kim (jykim33@wisc.edu), and Nate Richman (nrichman@wisc.edu)

CS 534 - Computational Photography
UW - Madison Fall 2018

Introduction

Biologists often study higher order organism functions by using *Model Organisms*, those which functionally display the same functions, but in a much smaller or simpler way. *Saccharomyces cerevisiae*, commonly known as Brewer's Yeast, is one of the simplest model organisms used in biology, however, the cellular mechanisms discovered in *S. cerevisiae* are the same processes that occur in human cells every day [1].

For this reason, *S. cerevisiae* yeast is still studied and used as a common Model Organism for genetic engineering, drug testing, and even further understanding intracellular processes. For example, *S. cerevisiae* is used as a model organism for developing the emerging technology of CRISPR/Cas9, which allows advanced genome editing [2].

Because of the widespread use of *S. cerevisiae* in biological research, it is important to have tools for high-throughput data collection of this yeast. One unique aspect of Brewer's Yeast is the clear visual distinctions throughout its life cycle [3]. This clear visual distinction has been a way for biologists to study phenotypic changes due to certain drugs and the effects of manipulating proteins involved in the mating response.

Currently, however, biologists have to sit at the microscope with a hand counter to identify cells in different stages, this is not conducive to high-throughput research. Secondly, counting by hand introduces extra experimenter bias that is not desired in biological research.

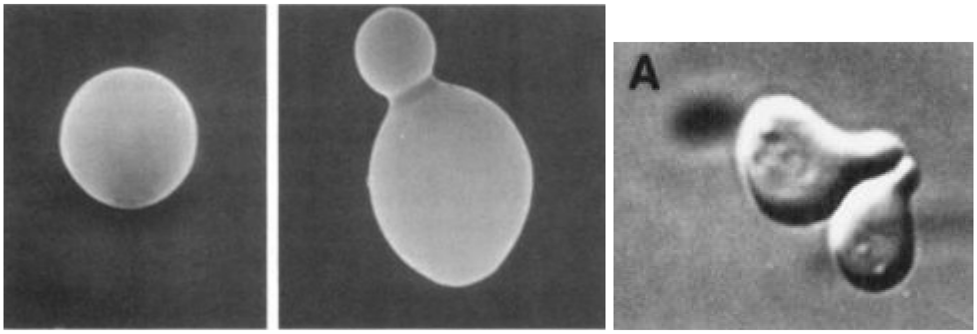

Figure 1: (left) Normal morphology of *S. cerevisiae* cell. (middle) Budding morphology of *S. cerevisiae* cell during mating [3]. (right) "Shmoo" morphology of *S. cerevisiae*, during the initial stage of mating after the sensing the other cell's pheromone [4].

Current state of the field

The field of image analysis for biology or microbiology is in no way new. Microbiologists have been trying to find computerized methods to automate tedious data collection and analysis since the 80s. Currently, there are multiple methods for cell/particle detection, and tracking with specific software, the most popular being ImageJ, developed by the Laboratory for Optical and Computational Instrumentation at the National Institutes of Health [5]. Other algorithms for cell counting have developed in MATLAB, using a process of thresholding images to black and white, and using connected component analysis to determine a full cell count [6]. Other algorithms have been able to output more significant data of the budding phenotype of *S. cerevisiae*. Laverty et al. demonstrated the use of a Cellometer software (cell counting) which would output circularity (roundness) of the detected objects in the picture, they were then able to place the objects into bins based on the roundness, effectively classifying budding phenotype [7]. The study was well done, however, it was not reported how the end-user would (or could) manipulate the object identification in the Cellometer software.

While ImageJ is fairly easy to use, an end-user who is not familiar with computational photography would not know how to analyze a photo or detect objects. ImageJ may work well for another developer who can write a plugin to detect cell shape, but for a microbiologist, it could not accomplish this solution without outside help. Similarly, the method presented Al-Khazraji et al. would not be able to distinguish between cell morphologies, which is the goal of this project [6]. And finally, while the method proposed by Laverty et al. would work for distinguishing between cell morphologies, the "black box" of the Cellometer is a large limitation of the approach [7]. Microbiologists need to be aware of any potential source of error or bias in their experiment, therefore, the algorithm needs to be "open" and tunable by the end user.

Proposed Solution

We are working on a re-implementation of the solution presented by Laverty et al., but with a few modifications. First, the cells used in Laverty's method do not form aggregates, whereas yeast cells often do, so our algorithm will need to be sensitive not only between the media and the cells, but also between the cells themselves. Second, Laverty's group only looked at dark-field microscopy (where the background is dark and the cells show up as light), our method will implement phenotypic detection in bright-field microscopy, where the cells show up as light on a light background. Third, the final implementation will be in MATLAB because MathWorks has student deals where MATLAB is generally free for research institutions, our end-user. And finally, our final UI will be easy to use and tunable, i.e. the experimenter will be able to set thresholds and bins to get a good reading that can then be used across their entire study and eliminate errors and sources of bias.

Current Progress and Difficulties

The strategy we are following is to determine cell boundaries, and translate it into a binary image. Next, we will fill in the boundaries to create connected components for each cell. We will then analyze the shape of each connected component, representing each cell, and determine the roundness of the cell. The method presented in Laverty et al. can be used to classify the individual cell into morphological bins, the program would then output the number of cells in each bin.

We have explored two methods of doing this so far, plain binary thresholding and operations on binary images, and using edge detection on the grayscale image.

Binary thresholding:

We have tried using the built-in graythresh method in MATLAB to determine an optimal threshold for converting a grayscale image to binary. This has been fairly successful in some images, however, some images have somewhat of a gradient from light to dark in the background, which means that a universal threshold cuts out certain cells. Thus, we may need to implement a local thresholding method or restrict our input images. When the thresholding has worked, we have been able to use thinning, and binary opening and closing on the image to find the cell edges, and then call regionprops to find the roundness of the cells, however many times, the cell aggregates are read as one single large region, so we need to find a way to differentiate between cells in a cell aggregate.

Edge detection:

We could either use the Sobel or Canny edge detection methods, initial testing has shown that the Canny edge detector is superior to the Sobel for this implementation. The Canny method seems to be more accurate since it relies on the second derivative, which can be evaluated when it reaches 0, instead of a local maximum or minimum, which may be fuzzy. Secondly, the Canny edge detector offers edge thresholding which will offer the user to set the threshold of whether a pixel is an edge.

A number of problems have come up from this method, first, the Canny method often detects multiple edges around the cells, this is due to the optical properties of the microscope used which has a lighter band around the edge of the cell, which shows up as darker, therefore the Canny edge detector picks up the edge between the background and the white band, and then the white band to the cell edge (figure 2). Secondly, during testing, our program often detects edges from noise inside the cells which increases the object count. So far, the fillholes function has remedied this problem but it is still clear that there are some objects with edges that aren't fully closed. We used watershed segmentation to address the issue. The histogram reveals the different amplitudes in gradient to find markers for the background and the desired objects. An estimate of the perimeter and area were obtained using the boundary coordinates of the image. The equation
$\frac{(4\pi A)}{P^2}$ returns a value less than one to determine roundness. Threshold values were set to determine which phase the *S. cerevisiae objects* are currently in.

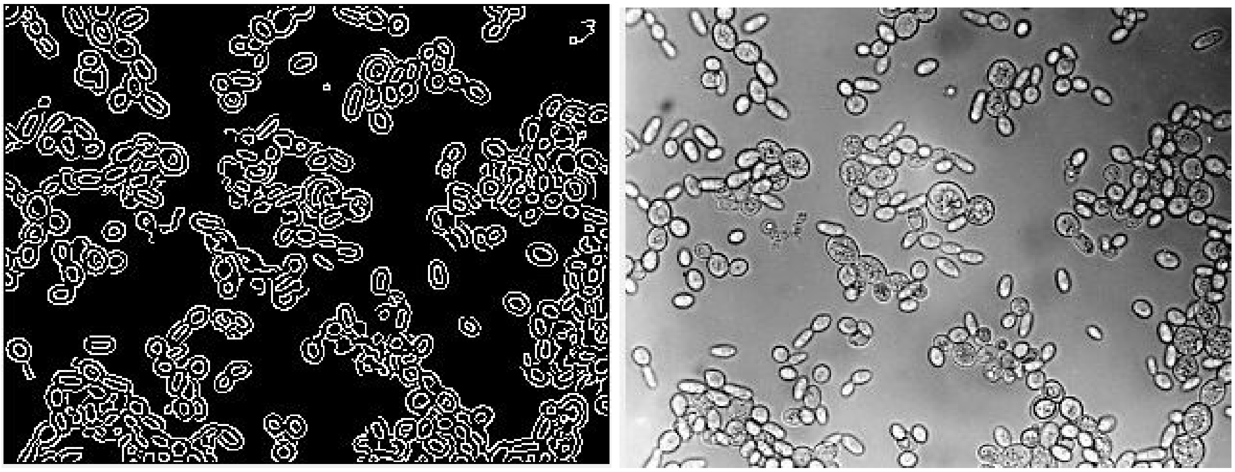

Figure 2: Illustration of the multiple edge problem in cells, binary edge detection using Canny edge detector on left, original image on right.

The problem we are currently facing is differentiating *S. cerevisiae* cells that are overlapping and budding, but we hope the watershed segmentation addresses the problem overlapping. Categorizing between Budding and "Shmoo" morphology is an ongoing concern.

Future Work and Conclusions

In order to evaluate whether our algorithm works and is accurate, we will be manually identifying the phenotypes present in multiple images. The algorithm most likely will have some limitations, such as possibly not being able to accurately determine the morphology of cells in a clump/cell aggregate, and not being able to identify the morphology, or even detect cells at the edges. While these may be limitations, they should not matter for the end user, because it will only matter that the results are consistent. In other words, the algorithm does not need to be 100%, but needs to be precise.

With sufficient testing, we should be able to determine whether the algorithm is precise. Finally, we want to test whether the UI is easy to use, and that there is enough that the user is able to manipulate so that the algorithm is less of a "black box" and more of a tool a researcher can use to consistently gather counts of cells in various life stages based on their morphology.

Timeline:

We would like to have a working algorithm done within the next couple weeks. As we mentioned above, we have a few possible starting points and ways to go forward at this point. We should be able to have a working algorithm by Saturday, November 10th. After this, we will continue to refine the algorithm by testing on images in the described manner. At the same time we can work on creating a UI with MATLAB's built-in GUIDE UI builder, or by using some of the other built UI builders in MATLAB. We will have the refined algorithm and the UI done by

Saturday, November 29th, and perform final testing as well as prepare the presentation before the presentations. As we go through, we will update the website, so it should be mostly done by the time of presentations.

Conclusions:

This project will give researchers and student researchers a useful tool that can save time, allow their process to be more high-throughput and reduce sources of error and biases within their experiments. This software could have large implications for the continuing research on *S. cerevisiae* by practiced researchers on topics such as cellular signaling and cytoskeletal responses. Both of these fields are current active research areas for understanding the pathology of cancer cells, as well as targets for cancer therapies. Not only could this software assist researchers in these areas, but it also has a large audience for use in teaching student-researchers. Since MATLAB is generally available for free to students, this will be an easy way for students to learn by doing research on *S. cerevesiae*, such as is done in the Biocore program at UW-Madison.

References

[1] D. Botstein, S. A. Chervitz, M. Cherry, "Yeast as a Model Organism," *Science*, vol. 277, no. 5330, p. 1259-1260, Aug. 1997.

[2] T. Jakociunas, I. Bonde, M. Herrgard, S. J. Harrison, M. Kristensen, et al., "Multiplex metabolic pathway engineering using CRISPR/Cas9 in *Saccharomyces cerevisiae*," *Metabolic Engineering*, vol. 28, p. 213-222, Mar. 2015.

[3] I. Herskowitz, "Life Cycle of the Budding Yeast *Saccharomyces cerevisiae*," *Microbiological Reviews*, vol. 52, no. 4, p. 536-553, Dec. 1988.

[4] J. Chenevert, N. Valtz, I. Herkowitz, "Identification of Genes Required for Normal Pheromone-Induced Cell Polarization in *Saccharomyces cerevisiae*," *Genetics*, vol. 136, no. 4, p. 1287-1296, Apr. 1994.

[5] National Institutes of Health, "ImageJ: Image Processing and Analysis in Java, [Online] Available: https://imagej.nih.gov/ij/index.html

[6] B. K. Al-Khazraji, P. J. Medeiros, N. M. Novielli, D. N. Jackson, "An automated cell-counting algorithm for fluorescently-stained cells in migration assays," *Biol. Proced. Online*, vol.13, no. 9, 2011

[7] D. J. Laverty, A. L. Kury, D. Kuksin, A. Pirani, et al., "Automated quantification of budding *Saccharomyces cerevisiae* using a novel image cytometry method," *J. Ind. Microbiol. Biotechnol.*, vol. 40, p. 581-588, 2013.