ERDC TR-19-21

Engineer Research and Development Center

US Army Corps of Engineers®
Engineer Research and Development Center

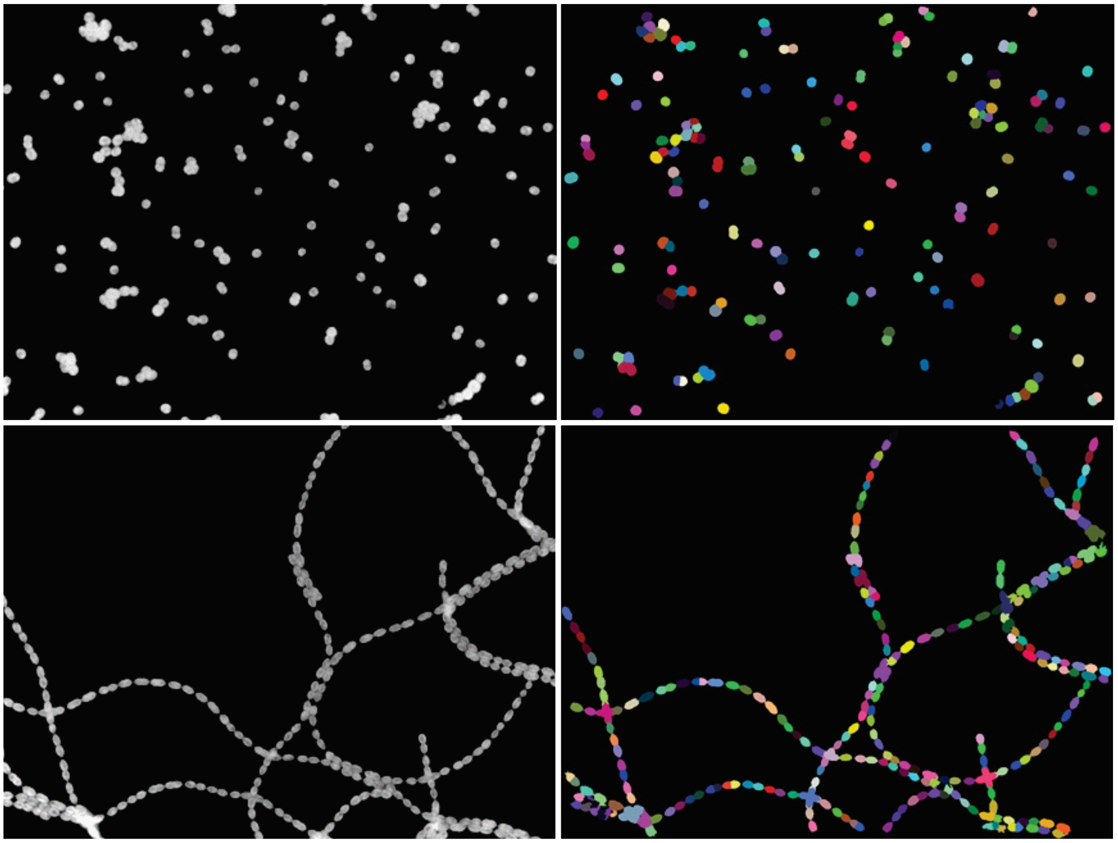
ERDC
ENGINEER RESEARCH & DEVELOPMENT CENTER

Aquatic Plant Control Research Program

A High-throughput Method for Counting Cyanobacteria Using Fluorescence Microscopy

Kaytee Pokrzywinski, Brandon Boyd, and Jarrell Smith

August 2019



ERDC TR-19-21
August 2019

A High-throughput Method for Counting Cyanobacteria using Fluorescence Microscopy

Kaytee Pokrzywinski

*U.S. Army Engineer Research and Development Center (ERDC)
Environmental Laboratory (EL)
Waterways Experiment Station, 3909 Halls Ferry Road
Vicksburg, MS 39180-6199*

Brandon Boyd and Jarrell Smith

*U.S. Army Engineer Research and Development Center (ERDC)
Coastal and Hydraulics Laboratory (CHL)
Waterways Experiment Station, 3909 Halls Ferry Road
Vicksburg, MS 39180-6199*

Final Report

Prepared for Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Project number 476547
Contract number W81EWF91309021
Program Element number U452127
Task number A1130

Abstract

Harmful algal blooms (HABs) occur all over the world naturally, but are increasing in frequency largely due to eutrophication. The U.S. Army Corps of Engineers (USACE) manages over 400 freshwater waterways making cyanobacteria HABs (cyanoHABs) a significant threat to water quality. CyanoHABs have been identified as a significant problem in USACE managed waterways where districts have reported associated fish kills and waterway closures. For USACE districts, when contracted out, the return time for cyanobacterial enumeration is approximately 30 days, therefore, there is a need to provide a better approach to counting cyanobacteria. However, counting cyanobacteria is notoriously difficult due to large filaments that make conventional counting methods impractical. It is anticipated that the application of this method for routine counting of field samples will greatly reduce analysis time, and therefore, will enhance the ability to combat HABs. In addition to cyanoHAB monitoring, cell densities are required for various down-stream techniques, including nucleic acid extractions where overloading of solid phase extraction columns can cause a loss of sample. This study developed automated image processing routines that enable rapid and accurate cell counting. The results of this method compared favorably with manual counting of single-cell and filamentous cyanobacteria cultures at a fraction of the analysis time.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Abstract ii

Figures and Tables iv

Preface v

Acronyms and Abbreviations vi

1 Introduction 1

1.1 Background 1
1.2 Objective 2
1.3 Approach 2
1.4 Purpose 2

2 Materials and Methods 4

2.1 Sample processing 4
2.2 Imaging 4
2.3 Manual image analysis 5
2.4 Automated image analysis 5
2.5 Enumeration 9

3 Results and Analysis 11

3.1 Analysis of single-celled cyanobacteria 11
3.1.1 Manual counting 11
3.1.2 Automated counting 12
3.1.3 Automated vs. manual counting of single cells 13
3.2 Analysis of filamentous cyanobacteria 14
3.2.1 Manual counting 14
3.2.2 Automated counting 15
3.2.3 Automated vs. manual counting of filaments 16
3.3 Limitations of the automated counting method 17
3.3.1 Image quality 17
3.3.2 Cell density 17
3.3.3 Quality assurance/quality considerations 22
3.3.4 Future improvements 22

4 Conclusions 23

References 25

Appendix A: Examples of analyzing images 26

Report Documentation Page

Figures and Tables

Figures

Figure 1. Raw microscopic image of *M. aeruginosa*. 6

Figure 2. Image of *M. aeruginosa* following pixel thresholding. 7

Figure 3. Image of *M. aeruginosa* with holes filled after thresholding and watershed filtering. 8

Figure 4. Color labels applied to image of *M. aeruginosa* after removal of cells touching the border. 9

Figure 5. Grayscale images of *M. aeruginosa* used for manual counting. Images contain 10% (A), 25% (B), 75% (C) and 100% (D) cell densities of M. aeruginosa. Note that D was not manually counted due to the occurrence of high-density cell clusters. 12

Figure 6. Comparison of processed images for 10% (A), 25% (B), 75% (C), and 100% (D) sample densities of *M. aeruginosa*. Each unique color represents a counted particle. 13

Figure 7. A comparison of automated and manual counting in *M. aeruginosa* samples at 10%, 25% and 75% of starting sample density. 14

Figure 8. Grayscale images of *Anabaena sp*. Sample A (left) and B (right) used for manual counting. 15

Figure 9. Example of processed images from automated method in *Anabaena sp*. Samples A (left) and B (right). Each unique color represents a counted particle. 16

Figure 10. A comparison of automated and manual counting in *Anabaena sp*. Samples A and B. 17

Figure 11. An example of poor image processing in both grayscale (A) and post processing (B). The method produces two types of errors, 1) it cannot recognize shapes of cells in large clusters and therefore truncates internal cells (right circle), and 2) it cannot differentiate cells within small clusters (left circle). 19

Figure 12. An example of overlapping filamentous cells in grayscale (A), and post processing (B). The white circles highlight a few areas where there were miss-counts due to overlapping filaments. 20

Figure 13. An example of overlapping filamentous cells in grayscale (A), and post processing (B). The white circles highlight a few areas where there were miss-counts due to overlapping filaments. 21

Tables

Table 1. Manual counts for a single *M. aeruginosa* sample at three concentrations. 11

Table 2. Automated counts for a single *M. aeruginosa* sample at four densities. 13

Table 3. Manual counts for *Anabaena sp*. Samples A and B. 15

Table 4. Automated counts for *Anabaena sp*. Samples A and B. 16

Preface

This study was prepared in response to the work-unit strategies for “Early Detection of Harmful Algal Blooms and Predicting Toxin Release: Linking Hyperspectral Imaging to Molecular Techniques.” This study was conducted for, and was funded by, the Aquatic Plant Control Research Program (APCRP) under project number 476547. Dr. Linda Nelson was the Program manager.

The work was performed by the U.S. Army Engineer Research and Development Center (ERDC), Environmental Laboratory (EL) and Coastal and Hydraulics Laboratory (CHL), Vicksburg, MS. At the time of publication, Dr. Brandon Lafferty was Chief, Environmental Processes (EP) Branch, Mr. Warren P. Lorentz was Chief, Environmental Processes and Engineering (EPE) Division, and Dr. Al Cofrancesco was the Technical Director (EZT). The Deputy Director of ERDC-EL was Dr. Jack E. Davis and the Director was Dr. Ilker R. Adiguzel.

COL Ivan P. Beckman was Commander of ERDC, and Dr. David W. Pittman was the Director.

Acronyms and Abbreviations

Acronym	Meaning
APCRP	Aquatic Plant Control Research Program
CHL	Coastal and Hydraulics Laboratory
cyanoHABs	cyanobacteria HABs
DoD	Department of Defense
EL	Environmental Laboratory
EP	Environmental Processes
EPE	Environmental Processes and Engineering
ERDC	Engineer Research and Development Center
FOV	field of view
HABs	Harmful algal blooms
JPEG	Joint Photographic Experts Group
mL	milliliters
mm	Millimeter
min	Minutes
nm	Nanometers
OD	Optical Density
PNG	Portable Network Graphics
TIFF	Tagged Image File Format
TR	Technical Report
TRITC/RFP	tetramethylrhodamine/red fluorescent protein
USACE	U.S. Army Corps of Engineers
W	Watt
µm	micrometer

1 Introduction

1.1 Background

Harmful algal blooms (HABs) occur all over the world naturally (Kim et al. 2008), however, due largely to increases in anthropogenic eutrophication, blooms are occurring more frequently, lasting longer, and are geographically expanding (Glibert et al. 2008; Anderson et al. 2012). Since the U.S. Army Corps of Engineers (USACE) manages upwards of 400 freshwater inland waterways, cyanobacteria, the most dominant type of blooming freshwater algae, pose a significant threat to their resources. All USACE managed waterways provide a variety of services including flood control, energy production, and navigation, as well as associated uses such as recreation, fish and wildlife management, and potable water supplies that are covered by the Clean Water Act (Brooks et al. 2015).

Cyanobacteria HABs (cyanoHABs) are a significant problem in USACE managed waterways where many district managers have reported negative impacts such as fish kills and waterway closures related to cyanoHAB events (Linkov et al. 2009). These events have an impact not only on operations, but also on recreational usage. In December 2016, the Great Lakes and Ohio River Division held a two day workshop on water quality and determined that there is incongruity in monitoring techniques used across districts. Additionally, there is often a significant delay in receiving information from contract laboratories once samples are sent for analysis. Furthermore, a 2014 USACE National HAB Survey, revealed that the majority of USACE districts characterize HABs through cell density, chlorophyll *a,* and/or toxin concentrations during bloom events.* Half of the surveyed districts intermittently monitor phytoplankton at the species level during the summer months, but the turnaround time for sampling and analysis is roughly 60–90 days for identification and enumeration. Therefore, there is a need for a more rapid, yet reliable method for the identification and enumeration of cyanobacteria for long-term monitoring by USACE districts. This method addresses the enumeration aspect of this

* Herman B, T. Clyde, E. Emery, B. Herman, G. A. Clyde Jr., E. Emery, J. Jung and K. Tackley. In Prep. Results of 2014 survey on harmful algal blooms within the Technical Report. Vicksburg, MS: U.S. Army Corps of Engineers.

problem and has the potential to be developed into an identification software.

1.2 Objective

The objective of this research was to develop a new, more efficient approach for counting filamentous and single-celled/colony forming cyanobacteria and to validate this approach against standard counting methods.

1.3 Approach

Similar approaches have been employed for enumerating aquatic bacteria (single cells) using fluorescent stains (Kepner and Pratt 1994; Seo et al. 2010; Stranka 2013; Boyer and Guevara 2015). This method, did not use fluorescent dyes, capitalizing instead on *in vivo* Chlorophyll *a* autofluorescence, something that is not possible in standard aquatic bacteria. Additionally, no existing methods have utilized an automated image processing approach using the freeware statistical program R, and no existing protocols have been adapted to operate on images containing filaments or clusters of cells. This study evaluated two cyanobacteria, one single-celled/colony forming *Microcystis aeruginosa*, and another filamentous cyanobacteria, *Anabaena* sp. These two species were chosen as they are the most dominant bloom forming cyanobacteria species within the continental United States and have substantially differing morphological features. This technical report (TR) centers on the development of a highly efficient, accurate and user-friendly method for counting cyanobacteria. Future iterations of this script can incorporate software interfaces avoiding the need for scripting and phytoplankton identification to, at a minimum, the class level.

1.4 Purpose

The purpose of this study was to identify a more efficient, user-friendly, and open-access method for counting cyanobacteria. Manual counting of images can be tedious and time consuming. Other methods for counting cells/particles in real-time (i.e., flow cytometry, the use of a hemocytometer, or Coulter counter) are not suited for filamentous cyanobacteria, as the flow cells/apertures/capillaries for these systems will often get clogged by filaments and colonies. Therefore, this TR provides valuable insights behind cyanobacteria cell counting using image processing in R for both novice and expert users of R/R-Studio. The ultimate goal is to develop an open-access pipeline and/or service for particle counting that produces similar results to conventional counting methods.

2 Materials and Methods

2.1 Sample processing

In this study, one to five milliliters (mL) of monocultures of the cyanobacteria *Microcystis aeruginosa* strain 2385 (University of Texas Culture Collection (UTEX)) and *Anabaena* sp strain 1448 (UTEX) were preserved to a final concentration of 1 % (*v*/*v*) glutaraldehyde at 4 °C for a minimum of one hour (up to one week) to crosslink membrane proteins and ensure morphological integrity when under vacuum. *M. aeruginosa* is a single celled and colony forming cyanobacteria where *Anabaena* sp. is a filamentous cyanobacteria.

The preserved sample was then added to an assembled 25 millimeter (mm) glass filter manifold containing a 0.2 micrometer (µm) black polycarbonate membrane on top of a 0.45 µm mixed cellulose ester backing filter. A low vacuum (< 5 mm Hg) was applied and the sample was filtered to dryness. The volume filtered was recorded, as this was necessary for the final density calculations. After filtration, the black polycarbonate membrane was removed from the manifold and placed facing up onto a clean microscope slide (25 mm x 75 mm standard thickness). Approximately 5 microliters (µL) of immersion oil was added to the center of the membrane and a glass coverslip was placed on top (25 x 25 mm squares). The coverslip was secured using clear nail polish at each corner. Processed samples were stored wrapped in foil at -20 °C until imaging.

2.2 Imaging

Previously frozen, prepared microscope slides were acclimated to room temperature prior to imaging. One drop of immersion oil type DF, a low-fluorescence oil used for fluorescence microscopy with very high resolution, was placed on top of the cover slip just prior to imaging. An Olympus BX53/43 manual epifluorescence microscope equipped with at 60x UPlanFL objective (1.25 Oil Iris Infinity/0.17 FN26.5 UIS2), 100 Watt (W) Hg and 75 W Xenon lamps with blue/green excitation (~480 nanometers (nm)) was used for imaging. The microscope was equipped with a tetramethylrhodamine/red fluorescent protein (TRITC/RFP) dichroic emission filter collecting light between 570 nm and 750 nm and a 10x focusing eyepiece. The TRITC/RFP filter was used to visualize *in vivo* Chlorophyll *a*, as opposed to a traditional fluorophore. The microscope was also outfitted with an Olympus DP80 monochrome camera. The camera adapter magnification was 0.63x, this is important as the camera field of view (FOV) is not the same as the eye piece. Images were acquired using cellSens v 1.13 (Olympus) with internal image size calibration. At least ten fields containing a minimum of roughly 30 cells per field of view were collected to achieve a final count of at least 300 cells.

For *M. aeruginosa,* a total of 40 images were collected at 100%, 75%, 25% and 10% (ten images per concentration) of the starting culture density to determine how effective the method was at estimating cell counts from sparse and highly dense samples. All 40 images of *M. aeruginosa* were subjected to both standard and automated counting methods.

For *Anabaena* sp. numerical counts were conducted on six independent samples at two cell densities, where each density was assessed in biological triplicates (30 images per sample, a total of 60 images for this species) using a similar cell density to the 25%–75% concentrations from the *M. aeruginosa* experiment. This was done to assess reproducibility of the method between samples.

2.3 Manual image analysis

Standard methods included manual counting by a user. Manual counting was conducted by one person prior to automated processing as to avoid bias. The amount of time to manually count ten images (per sample) was also monitored.

2.4 Automated image analysis

The method presented in this TR describes the sequence of image processing steps required to obtain an automated count of cyanobacteria using a custom script. A detailed example using annotated source code can be seen in Appendix A. Analyses for automated image processing were conducted using the freeware statistical programs R, R-Studio, and Bioconductor. The primary package used within Bioconductor was EBImage (Pau et al. 2010).

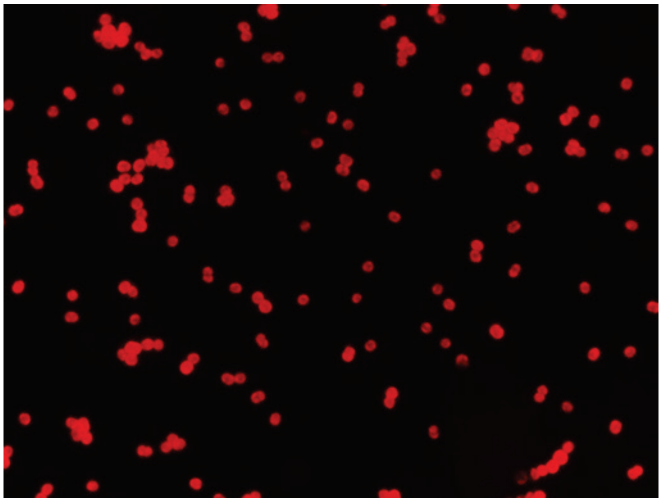
> “EBImage provides general purpose functionality for image processing and analysis, particularly in the context of high-throughput microscopy-based cellular assays, EBImage offers tools to segment cells and extract

> quantitative cellular descriptors. This allows the automation of such tasks using the R programming language and facilitates the use of other tools in the R environment for signal processing, statistical modeling, machine learning and visualization with image data."

Other packages used for image manipulation and reading within R include tiff, pixmap, and rtiff. This counting method was similar in approach to other standard particle counting/tracking programs (e.g., ImageJ) (Particle Analysis 2017) but allows for automated adaptive thresholding and filtering of each image without the need for manual adjustment using previously validated algorithms.

1. Raw microscopic images were first uploaded into R/R-Studio as tiff files (Figure 1). Other compatible file formats include joint photographic experts group (JPEG) and portable network graphics (PNG).

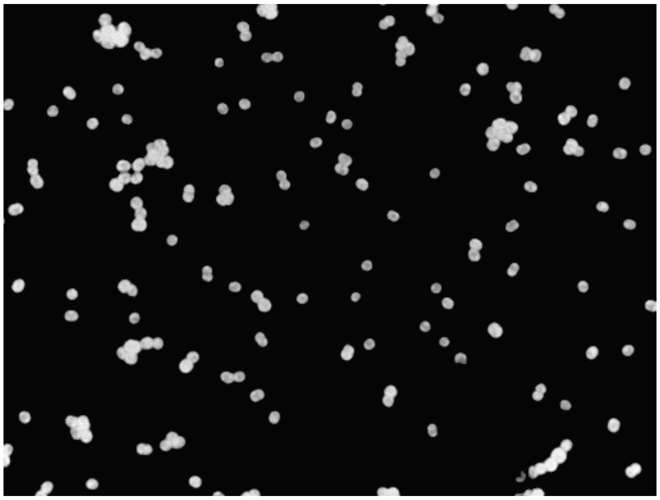
Figure 1. Raw microscopic image of *M. aeruginosa*.



2. The first processing step was pixel thresholding, which is the simplest method of image segmentation. Thresholding methods replace each pixel intensity below a user-specified fixed constant in an image with a black pixel at a value of 0. In this case, the pixel threshold was empirically tested and set at an intensity 0.3 for the red, green, and blue (RGB) channels. This threshold was used to separate adjacent particles and reduce background noise without omitting important information as was required for accurate counting in downstream processes. In this step, an image was first converted to a pixel intensity map as a matrix (where *x* and *y* coordinates corresponded with image locations) and edited numerically to

replace pixels with intensities less than 0.3. The pixel maps were then converted back to a single channel image file in grayscale form (Figure 2).

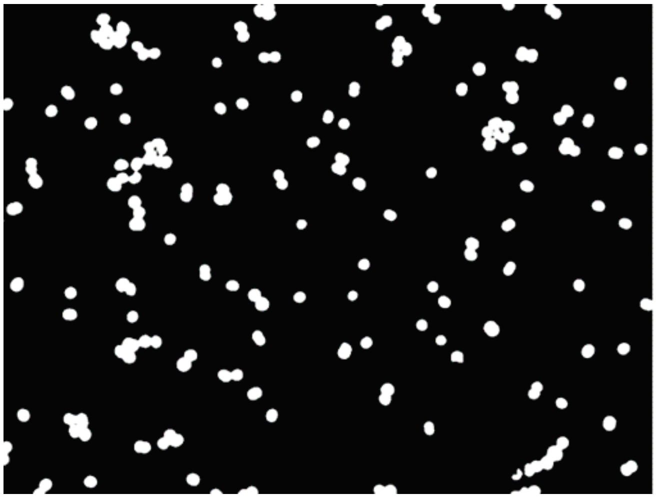
Figure 2. Image of *M. aeruginosa* following pixel thresholding.



3. Next, an adaptive image threshold was applied to further segment the image using the function *thresh* in EBImage. This type of thresholding is clustering-based and assumes that the distribution of image pixel intensities follows a bimodal histogram, and separates those pixels into two classes, a foreground and background, ultimately creating a binary image (Figure 3). Using this approach, the threshold value can be calculated for each image frame separately, independent of the fluorescence intensity of each frame. This allows for the successful use of an endogenous marker, such as Chlorophyll *a,* where fluorescence intensities can vary by cell type, general health, and laser intensity/drift.
4. From here, a watershed filter was used to segment connected objects within the binary image using the function *watershed* in EBImage. A watershed filter treats an image like a topographic map, finding the edges of particles between different intensities. Therefore, this filter identified and separated objects that were different than the background, and hence, works well on binary images. Although there were no visible differences between images (Figure 3), this was required for accurate enumeration in later steps.
5. Next, holes within particles caused by thresholding and watershed filtering needed to be filled so as to avoid bias during particle counting in subsequent steps. To accomplish this, the function *fillHull* was used in EBImage. This is particularly important when chloroplast-like organelles are in the cell periphery, leaving a less intense hole in the center of the cell after pixel and image thresholding. Unfilled particles would give the

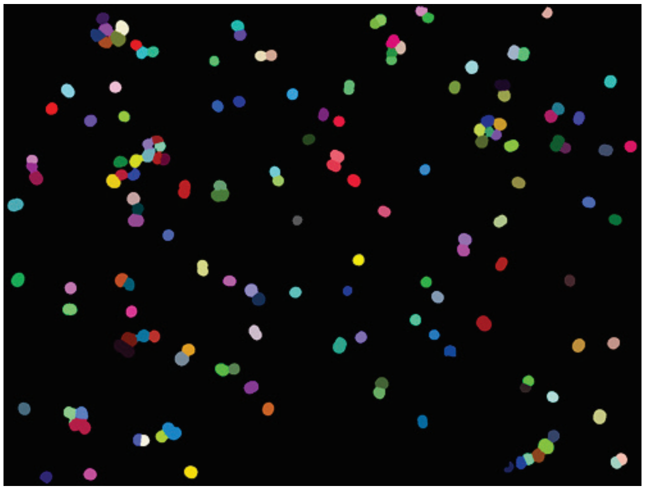
appearance of doughnuts and need to be compensated for before counting. This step was more an insurance step so that counting proceeded properly in the following steps. In binary form, unique objects were identified as sets of pixels with the same unique integer value (i.e., 1 (white) completely surrounding a different integer value, 0 (black) and filled-in with the appropriate value) (Figure 3).

Figure 3. Image of *M. aeruginosa* with holes filled after thresholding and watershed filtering.



6. The final processing step performed in this study was to remove cells touching the edges as to avoid over estimation between frames/image areas; this prevents situations where the same cell could be counted multiple times across frames. Initially, the border of the image was detected using the *dims* function in the base R package. Then pixels that touched the edge/border of the image having values above 0 were identified and removed using the *unique* and *rmObjects* functions in the base R package. Finally, color coding was applied the total number of unique particles were counted using the max function in the base package in R.

Figure 4. Color labels applied to image of *M. aeruginosa* after removal of cells touching the border.



7. Image objects (i.e., cyanobacteria cell) were then color coded as a quick visual confirmation that the image segmentation algorithms appropriately distinguished clusters/filaments of cells. At random, each object was assigned a unique color using the function *colorLabels* in the package EBImage (Figure 4). Adjacent cells having the same color did not separate well during the segmentation process, and would be considered one object during enumeration.
8. After removing the cells on the edges, a color label was again applied and the particles in the new image were counted (Figure 5). Images were counted by identifying the number of unique objects (i.e., pixels having values greater than 0 and not touching each other). Results from all ten images for each sample were output into a table format for averaging and back calculating the number of cells per mL. The amount of time to automate the counting of ten images (per sample) was also monitored during these methods, including path/directory creation and file transport.

2.5 Enumeration

Final cyanobacterial densities for each species were calculated using the following equations modified (from Wetzel and Likens 1991).

$$\text{Field of view (FOV)} = H\,W - 2\,D * (H + W) + 4\,D^2 \tag{1}$$

$$\text{Cyanobacteria mL}^{-1} = (MCF * N) \,/\, \text{Volume Filtered} \tag{2}$$

$$\text{Total Cyanobacteria} = \text{Cyanobacteria mL}^{-1} * \text{Total Volume} \tag{3}$$

Where, *MCF* = Membrane Conversion Factor = Filtration area / FOV, *N* = Total number of cyanobacteria counted / number of fields counted, *VF* = Volume filtered, *TV* = Total sample volume, Calibrated Image Size = Image size = pixels, Calibration factor = µm /pixel, Calibrated size = Image size *x* calibration factor, *H* = Image height (µm), *W* = Image width (µm), *D* = Median cell diameter (µm).

Example:

Filtration area = Funnel diameter = 25 mm
Volume filtered = 1 mL
Total sample volume = 100 mL
H = 207.5 µm
W = 155.6 µm
D = 2.5 µm

FOV = 207.5 * 155.6 – 2 * 2.5 * (207.5 + 155.6) + 4 * 2.5^2 = 30495.6 μm^2
MCF = 491 mm^2 / 0.0305 mm^2 = 16098
N = 2478 cells / 10 fields = 248 cells/field
Cyanobacteria mL^{-1} = (16098 * 248 cells) / 1mL = 3.99×10^6 cells mL^{-1}
Total Cyanobacteria = 3.99×10^8 cells

3 Results and Analysis

3.1 Analysis of single-celled cyanobacteria

3.1.1 Manual counting

Manual counting took approximately 1–5 minutes (min) for each *M. aeruginosa* image depending on cell density (1–5 min *10 images = 10–50 min/sample). Manual counting of the 75% concentration was challenging owing to the occurrence of small clusters, nevertheless, counting was still attempted for this concentration. However, the 100% concentration was not manually counted since there was significant clustering between adjacent cells that made visible separation of particles challenging. Therefore, manual counting may not be practical, depending on the cell density and time available to analyze the images. It also may not be practical when dealing with large datasets. The actual counts and estimated time for counting the 10%, 25%, and 75% concentrations can be found in Table 1.

Table 1. Manual counts for a single *M. aeruginosa* sample at three concentrations.

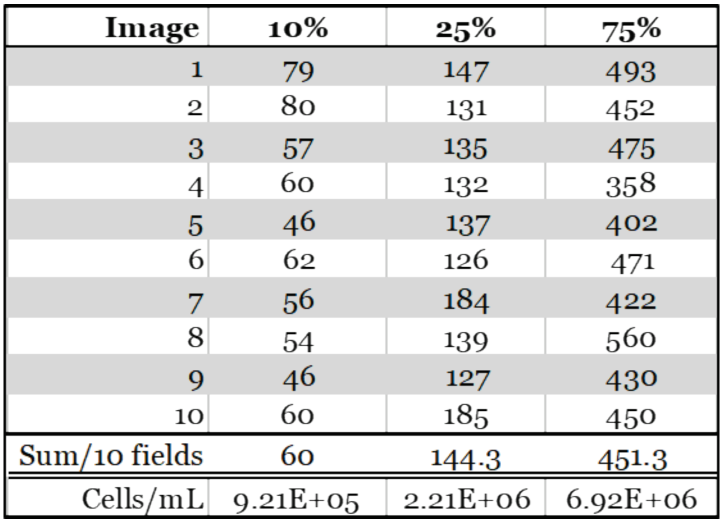
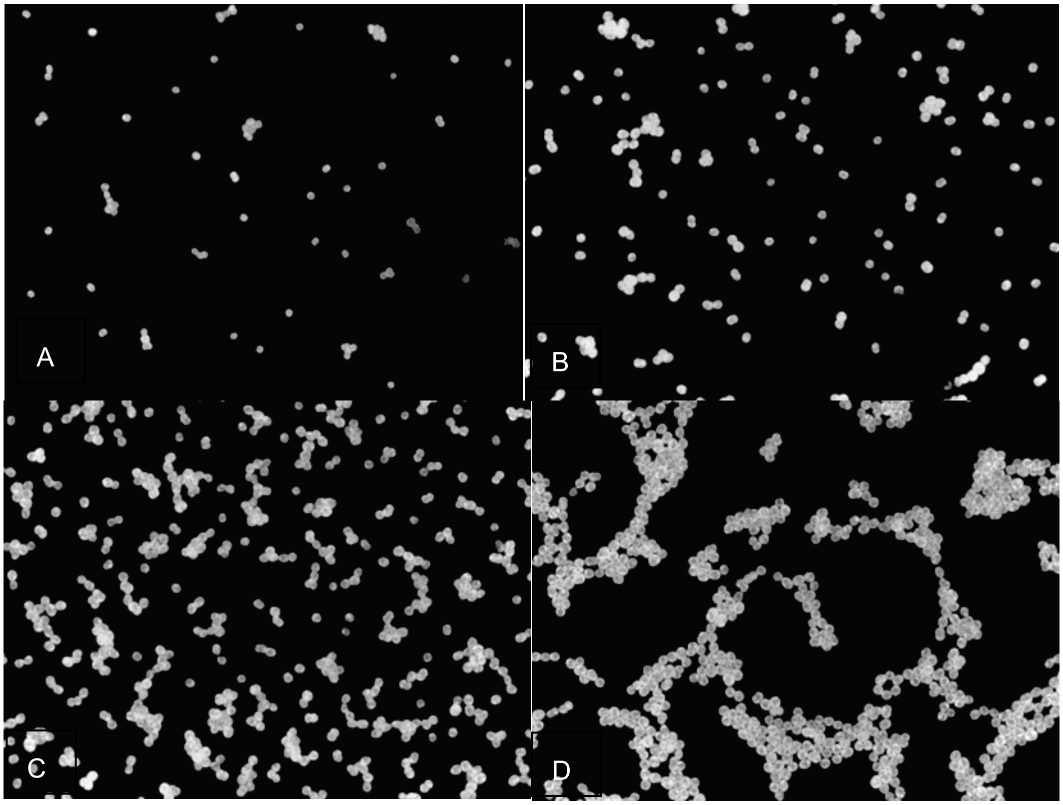


Image	10%	25%	75%
1	79	147	493
2	80	131	452
3	57	135	475
4	60	132	358
5	46	137	402
6	62	126	471
7	56	184	422
8	54	139	560
9	46	127	430
10	60	185	450
Sum/10 fields	60	144.3	451.3
Cells/mL	9.21E+05	2.21E+06	6.92E+06

There was a considerable difference in the total detected fluorescence in the raw images acquired from the microscope. However, prior to the grayscale conversion, the image thresholds were adjusted to account for these natural fluctuations (enabling better quality images for publication). Additionally, the method was able to accurately determine cell density independent of fluorescence intensity so no additional bias was introduced by adjusting thresholds; see Appendix A for unadjusted raw images (Figure 5).

Figure 5. Grayscale images of *M. aeruginosa* used for manual counting. Images contain 10% (A), 25% (B), 75% (C) and 100% (D) cell densities of M. aeruginosa. Note that D was not manually counted due to the occurrence of high-density cell clusters.



3.1.2 Automated counting

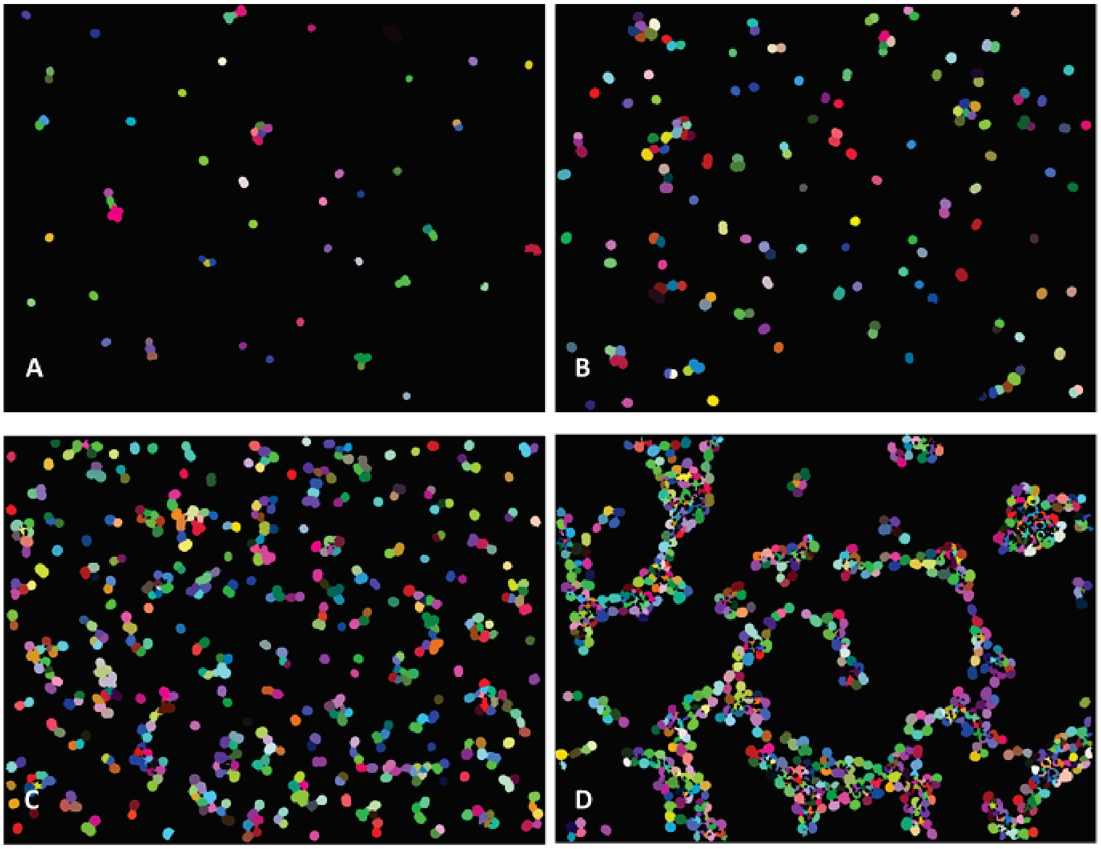
Collected images were analyzed using the automated method detailed in Section 2.4. Example raw and masked images can be seen in Appendix A. Additionally, examples of processed images of each density evaluated can be seen in Figure 6. The automated counting method was able to accurately identify and enumerate single cells and cells within small clusters (Figure 6 A–C), however, it was not effective at accurately detecting cells within large clusters or in high-density samples (Figure 6D).

It took approximately 2.5 min to count each sample (ten images); most of this time involved actively moving processed images/files to permanent directories.

Table 2. Automated counts for a single *M. aeruginosa* sample at four densities.

Image	10%	25%	75%	100%
1	70	154	483	880
2	67	145	442	877
3	51	134	466	838
4	52	142	376	844
5	46	129	400	791
6	56	132	459	838
7	63	171	424	813
8	59	139	451	770
9	47	122	426	684
10	55	171	437	641
Sum/10 fields	56.6	143.9	436.4	797.6
Cells/mL	9.11E+05	2.32E+06	7.03E+06	1.28E+07

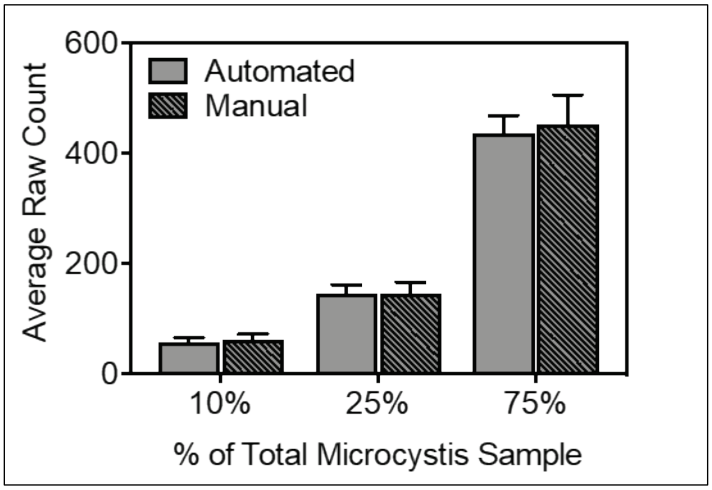
Figure 6. Comparison of processed images for 10% (A), 25% (B), 75% (C), and 100% (D) sample densities of *M. aeruginosa*. Each unique color represents a counted particle.



3.1.3 Automated vs. manual counting of single cells

A comparison between the automated and manual counting was performed to assess reliability of the new counting technique on single celled cyanobacteria and identify methodological improvements. As seen in Figure 7, there is exceptional agreement between manual and automated counting strategies. To confirm there were no significant differences between the two methods, a two-tailed, paired t-test was performed using the program Graphpad PRISM v 7.02 and the resulting p-value was non-significant at 0.1174.

Figure 7. A comparison of automated and manual counting in *M. aeruginosa* samples at 10%, 25% and 75% of starting sample density.

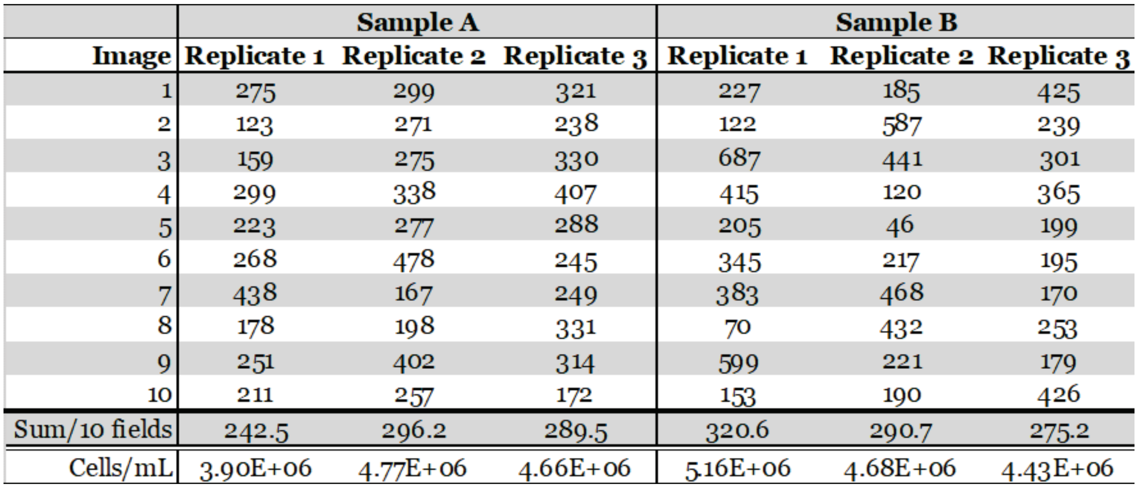


3.2 Analysis of filamentous cyanobacteria

3.2.1 Manual counting

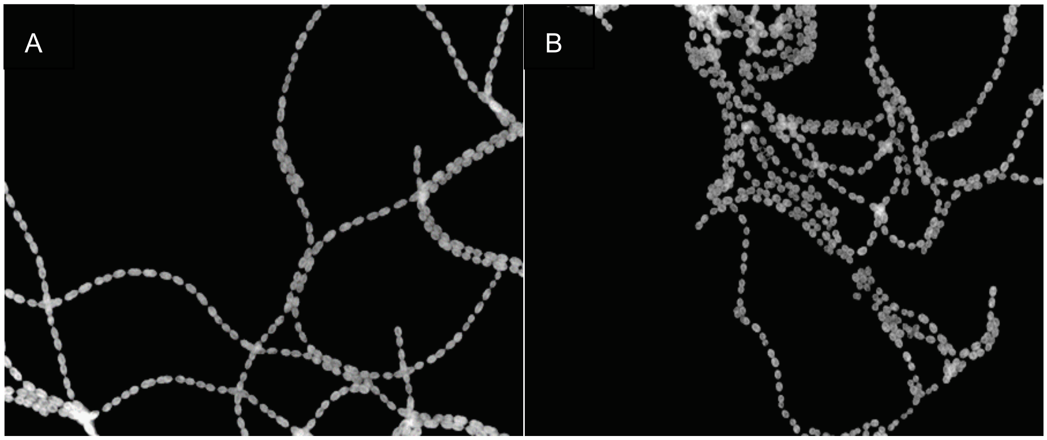
Manual enumeration took approximately 3–5 min to count each image (3-5 min * 10 images = 30–50 min = per sample), meaning it took approximately 3–5 hours to count 2 samples in biological triplicates (60 images) (Table 3). Manual counting of filamentous samples may also not be practical when working with large datasets or analyzing experiments that consist of at least two treatments in biological triplicates over multiple days. The actual counts for six independent *Anabaena* sp. samples (two conditions Sample A and Sample B in triplicate) can be found in Table 3.

Table 3. Manual counts for *Anabaena sp.* Samples A and B.



	Sample A			Sample B		
Image	Replicate 1	Replicate 2	Replicate 3	Replicate 1	Replicate 2	Replicate 3
1	275	299	321	227	185	425
2	123	271	238	122	587	239
3	159	275	330	687	441	301
4	299	338	407	415	120	365
5	223	277	288	205	46	199
6	268	478	245	345	217	195
7	438	167	249	383	468	170
8	178	198	331	70	432	253
9	251	402	314	599	221	179
10	211	257	172	153	190	426
Sum/10 fields	242.5	296.2	289.5	320.6	290.7	275.2
Cells/mL	3.90E+06	4.77E+06	4.66E+06	5.16E+06	4.68E+06	4.43E+06

Figure 8. Grayscale images of *Anabaena sp.* Sample A (left) and B (right) used for manual counting.



3.2.2 Automated counting

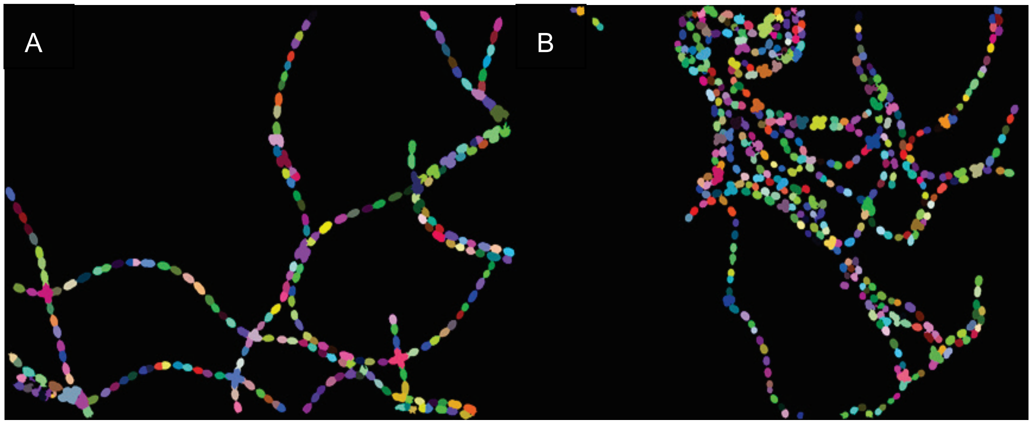
As in the single-cell detection method, collected images were enumerated using the automated analysis detailed in Section 2.4. Example raw and masked images can also be seen in Appendix A. Additionally, examples of processed images of each density evaluated can be seen in Figure 9. The automated counting method was able to accurately identify and enumerate single layers of filaments (Figure 9A), however, it was not effective at accurately detecting cells in overlapping filaments (Figure 9B). This challenge was not unique to the automation process, but was also an issue in the manual counting method.

Automated counting took approximately 2.5 min per sample (10 images); where again, most of this time involved actively moving processed images/files to permanent directories.

Table 4. Automated counts for *Anabaena sp*. Samples A and B.

	Sample A			Sample B		
Image	Replicate 1	Replicate 2	Replicate 3	Replicate 1	Replicate 2	Replicate 3
1	275	307	334	205	202	414
2	251	249	254	107	575	260
3	211	258	322	694	456	296
4	301	326	403	425	123	357
5	293	286	300	201	48	229
6	298	499	222	361	226	200
7	217	177	244	394	459	158
8	219	180	361	72	440	258
9	214	413	298	615	239	186
10	299	236	152	151	200	443
Sum/10 fields	257.8	293.1	289	322.5	296.8	280.1
Cells/mL	4.15E+06	4.72E+06	4.65E+06	5.19E+06	4.78E+06	4.51E+06

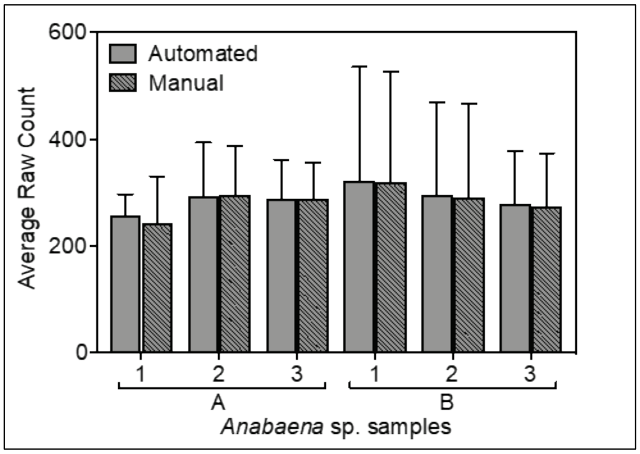
Figure 9. Example of processed images from automated method in *Anabaena sp*. Samples A (left) and B (right). Each unique color represents a counted particle.



3.2.3 Automated vs. manual counting of filaments

A comparison between the automated method and manual counting was performed to assess reliability of the new counting technique on filamentous cyanobacteria. Similar to the *M. aeruginosa* analysis, there is excellent agreement between manual and automated counting strategies (Figure 13). To confirm there were no significant differences between the two methods, a two-tailed, paired t-test was performed and the resulting p-value was non-significant at 0.4287.

Figure 10. A comparison of automated and manual counting in *Anabaena sp.* Samples A and B.



3.3 Limitations of the automated counting method

3.3.1 Image quality

This method provided reliable images for both manual and automated cell counts. Using either method, image quality is critical for proper enumeration. As a general precaution, issues may arise if uploading poor quality images or for images that are blurred. Images containing non-specific particles are of less concern (unless they physically block algal/cyanobacterial cells or are fluorescent in the same excitation/emission range as Chlorophyll *a*), as the program will not be able to differentiate a cell from other erroneous fluorescent particles or fibers. It is important to note that only particles that naturally fluoresce using an excitation/emission similar to Chlorophyll *a* will be observed using this method, therefore, sediment and other non-algal/cyanobacterial particles from field samples will be ignored. Typically, even if non-specific particles are fluorescent, they can be removed via thresholding considering that algal/cyanobacterial autofluorescence is typically high. To help with blurred images or a halo effect, it is important to ensure that an appropriate cover slip is used, immersion oil and the appropriate objective when imaging, and that samples are free of non-specific debris to produce the highest quality image for automated counting. Pre-filtration at higher pore sizes may be required to remove larger detritus or non-specific particles.

3.3.2 Cell density

Sample dilution can improve issues inherent to both the manual and automated counting methods by reducing the occurrence of overlapping cells and large clusters. High density particles are problematic as the program cannot define cell edges, and therefore, improperly calls/codes particles (Figure 11). The lack of round/regular patterns within clusters and high frequency of colors found within, not linked to, a typical cellular shape is characteristic of errors in large clusters using this program/pipeline. Additionally, situations where there are overlap of cells should be avoided, as the program cannot accurately differentiate the area of overlap, and often in filamentous cyanobacteria, four cells may be identified as a single large particle (Figure 12). Although a few overlapping cells won't dramatically influence the accuracy of the cell counts, these situations should be avoided. Realistically, overlapping cells are also difficult to count/interpret when manually counting as well, and as such, these situations should also be avoided. However, this program may be no more error prone when counting overlapping cells as the human eye. It is therefore suggested to use a cell density less than 1x 10^7 cells/mL, but greater than 1x10^5 cells/mL, to maintain at least 30 particles per frame to ensure the most accurate counting when performing the automated method. The actual cell density may vary based on the type/morphology (i.e., filaments, colonies or single cells) and size of cells used during this analysis.

Figure 11. An example of poor image processing in both grayscale (A) and post processing (B). The method produces two types of errors, 1) it cannot recognize shapes of cells in large clusters and therefore truncates internal cells (right circle), and 2) it cannot differentiate cells within small clusters (left circle).

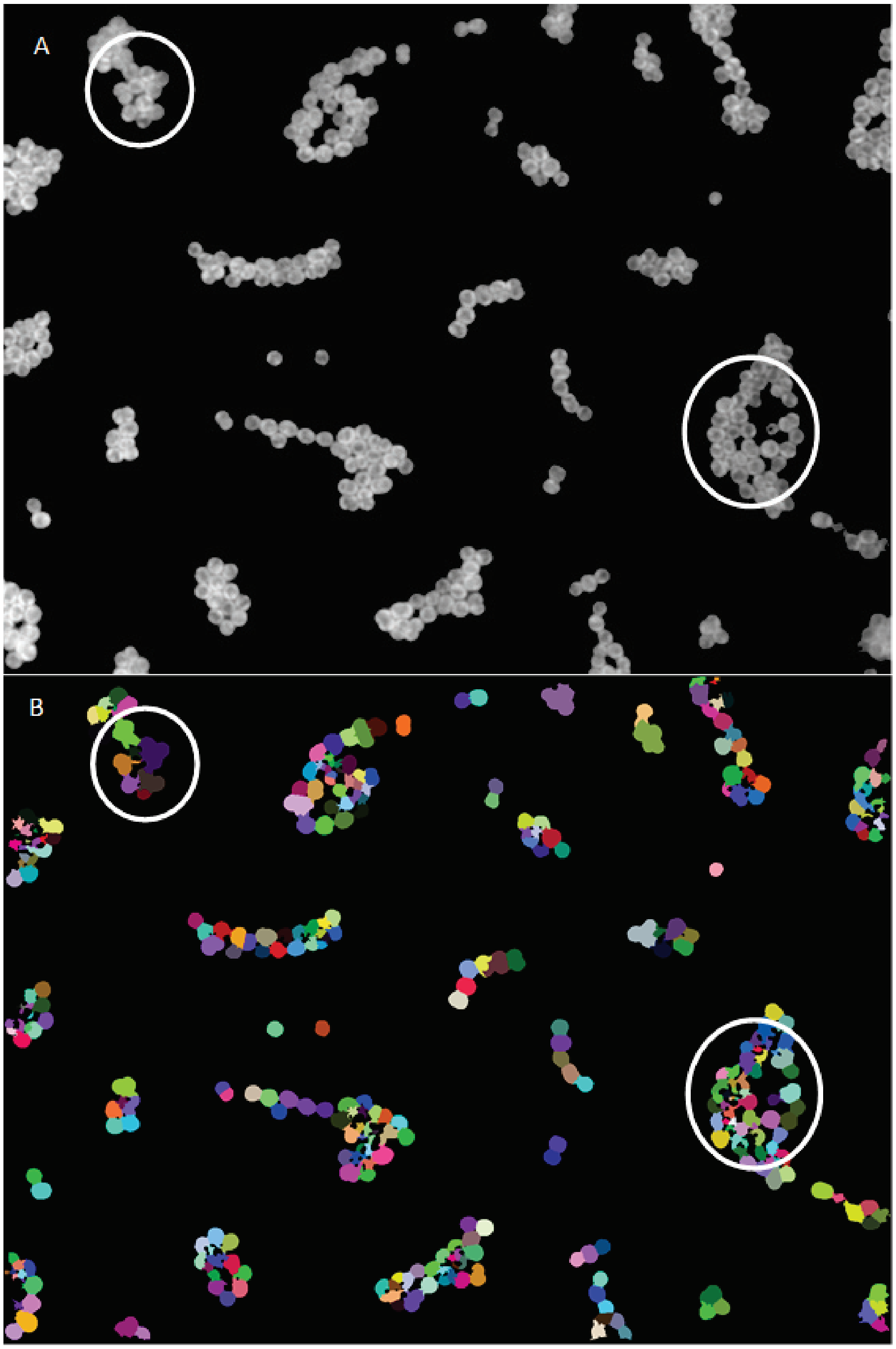


Figure 12. An example of overlapping filamentous cells in grayscale (A), and post processing (B). The white circles highlight a few areas where there were miss-counts due to overlapping filaments.

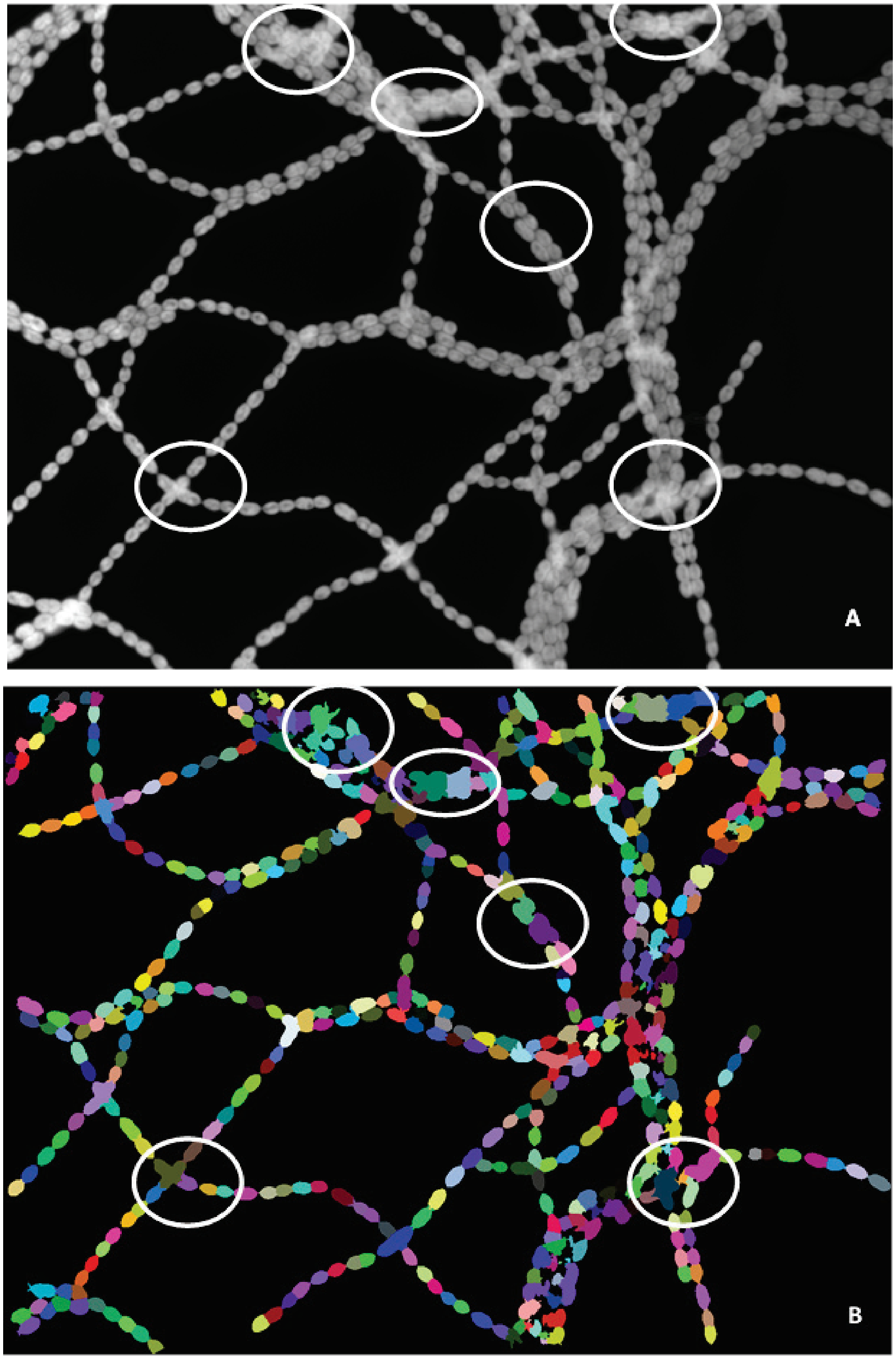
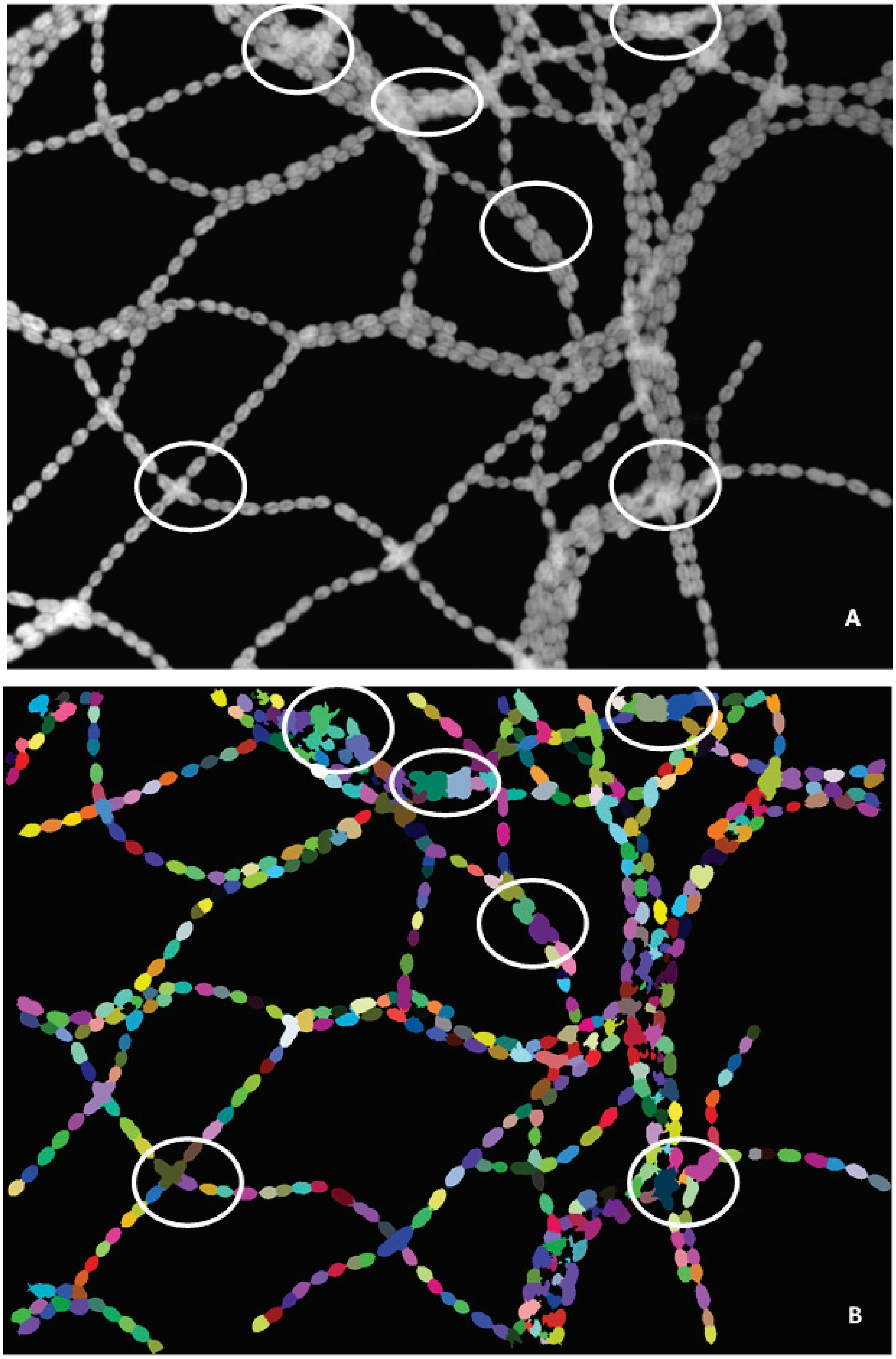


Figure 13. An example of overlapping filamentous cells in grayscale (A), and post processing (B). The white circles highlight a few areas where there were miss-counts due to overlapping filaments.



3.3.3 Quality assurance/quality considerations

This method provides an excellent quantitative metric for cell density that is highly comparable to manual counting in substantially less time. However, it is important to understand the limitations and benefits of using such a method for this application including image quality, sample type/morphology, and total density/need for sample dilution. It is therefore recommended that an optical density (OD) threshold be established as a guide for sample dilution so as to avoid overcrowding and overlapping situations when imaging and also to ensure there are at least 30 cells per field of view.

3.3.4 Future improvements

There are many read/write steps in the script and images, in particular, tagged image file format (TIFF) are especially slow to write. The method can be improved by editing the script so to remove image read/write during intermediate steps or to store them to memory instead of disk, which could ultimately reduce runtimes from minutes to seconds. Additionally, it is possible to input strategies for cell type classifications into the code, including size, shape, extrusions, internal complexity and color among others, to identify and reject non-target particles and for rapid total phytoplankton counts. Future iterations will work to incorporate more algal class specifications for rapid algal identification and enumeration.

4 Conclusions

This report compares the use of a newly developed automated image processing method using R/R-studio with manual counting for both single-celled (*M. aeruginosa*) and filamentous (*Anabaena* sp.) cyanobacteria. The major finding is that the automated enumeration method performs comparably with the manual method in significantly less time (approximately 1/12 to 1/20 of the time). The automated enumeration method is therefore likely to save time and money in determining cyanobacterial densities on at least an order of magnitude, based on analysis time in this study. The development of this method was critical for assessing cell densities required for optimal nucleic acid extractions to be used in rapid molecular screening of cyanobacterial genera and cyanotoxin genes. The results of which could be used to develop a direct relationship (i.e., standard curve) with optical density for each genera or species of cyanobacteria in culture, to provide a suitable approximation of cyanobacteria density.

Although this method was performed and validated in laboratory monocultures cultures of cyanobacteria, future iterations of this method would focus on enumeration in mixed algal/cyanobacterial/bacterial systems with total cell densities in the range of $1x10^5$ to $1x10^7$. Given that all algae auto-fluoresce, and other fluorometric stains such as DAPI have been used to enumerate bacteria using different semi-automated counting methods, it is highly likely that this method would successfully work to enumerate field samples of algae/cyanobacteria/bacteria with minor modifications, under variable magnifications and/or when using different and even multiple fluorometric dyes. Furthermore, a unique emission filter for phycocyanin could be used in addition to a Chlorophyll *a* emission filter to determine the relative ratio of cyanobacteria in mixed phytoplankton communities.

Several limitations exist for this technology that are similar in practice to the limitations of manual cell counting. Given the increased speed and objectivity of the automated counting method, the limitations were considered acceptable. However, future improvements can be made to overcome limitations of the method. For instance, sample dilution can be used to avoid and reduce overcrowding and overlapping cells. Furthermore, the script generated in this study can form the foundation of an algal classification module in future studies. This future module can work to enable rapid total algal enumeration and identification at higher taxonomic levels (i.e., class level), based on overall size and pigment types. This method is similar in approach to other methods that use nucleic acid stains and particle counting programs such as ImageJ, however, the use of R to automate and simplify this procedure significantly reduced processing time in a fairly user-friendly, free platform. Future iterations could include a GUI interface for less-experienced programming users.

The current analysis time for USACE district customers is 60–90 days for algal/cyanobacterial enumeration and identification. It is estimated that resource managers who use this method could greatly decrease analysis turn-around-time (< 1–2 days), and therefore, enhance their ability to combat HABs. Although this method does not allow for identification, total cell density can be used to guide management of algal blooms, and in the script, could be modified so that dual filters could be used to determine a relative ratio or percent contribution of cyanobacteria to a given community structure. Although not specific to cyanobacteria, the application of this method for routine counting of phytoplankton in field samples would assist resource managers in anticipating HAB events, thereby, providing information during the pre- or early bloom stages where management tactics are typically more successful. This study forms the foundation for future automated counting methods that would ultimately rely on machine learning and facial recognition software to rapidly (< 1–2 days) identify unique algal classes. It is an important step towards the rapid identification of cyanobacteria from eukaryotic algae, and therefore towards early detection and rapid response strategies for the effective management of freshwater HABs.

References

Anderson, D. M., A. D. Cembella, and G. M. Hallegraeff. 2012. Progress in understanding harmful algal blooms: Paradigm shifts and new technologies for research, monitoring, and management. *Annual Review of Marine Science* 4:143–176. https://doi.org/10.1146/annurev-marine-120308-081121

Boyer J, Guevara R. 2015. Bacteria enumeration by epifluorescence microscopy using DAPI stain. Florida Coastal Everglades LTER. http://fcelter.fiu.edu/data/protocols/Bacterial_Enumeration_protocol.pdf

Brooks, B. W., J. M. Lazorchak, M. D. A. Howard, M-V. V. Johnson, S. S. L. Morton, D. A. K. Perkins, E. D. Reavie, G. I. Scott, S. A. Smith, and J. A. Steevens. 2015. Are harmful algal blooms becoming the greatest inland water quality threat to public health and aquatic ecosystems? *Environmental Toxicology and Chemistry* 35:6–13. https://doi.org/10.1002/etc.3220.

Glibert, P. M., E. Mayorga, and S. Seitzinger. 2008. *Prorocentrum minimum* tracks anthropogenic nitrogen and phosphorus inputs on a global basis: Application of spatially explicit nutrient export models *Harmful Algae* 8(1):33–38. https://doi.org/10.1016/j.hal.2008.08.023.

ImageJ. 2017. Particle analysis. Madison, WI: University of Wisconsin-Madison. https://imagej.net/Particle_Analysis.

Kepner, R., and J. Pratt. 1994. Use of fluorochromes for direct enumeration of total bacteria in environmental samples: Past and present. Microbiological Reviews. 58(4):603–615.

Kim, M., S-Y. Jeong, and S-J. Lee S. 2008. Isolation, identification, and algicidal activity of marine bacteria against *Cochlodinium polykrikoides*. *Journal of Applied Phycology* 20(6): 1069–78.

Linkov I, Satterstrom F, Loney D, Steevens J. 2009. *The Impact of Harmful Algal Blooms on USACE Operations*. ERDC/TN ANSRP-09-1. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Pau, G., F. Fuchs, O. Sklyar, M. Boutros, and W. Huber. 2010. EBImage—an R package for image processing with applications to cellular phenotypes. *Bioinformatics* 26(7):979–981. https://doi.org/10.1093/bioinformatics/btq046.

Seo, E, T. Ahn, and Y. Zo. 2010. Agreement, precision, and accuracy of epifluorescence microscopy methods for enumeration of total bacterial numbers. *Applied Environmental Microbiology* 76(6):1981–1991.

Stranka I. 2013. Counting bacteria and viruses with ImageJ. https://www.researchgate.net/file.PostFileLoader.html?id=563b0c4d6225ffe2cc8b45bc&assetKey=AS%3A292338548133889%401446710349732

Appendix A: Examples of analyzing images

Prior to analyzing images, install the free statistical programs R and R-studio available at: https://www.r-project.org/ and https://www.rstudio.com/. This code was constructed under R v 3.4.1 and R-studio v 1.0.153. This method uses Bioconductor and packages contained within. Bioconductor is an open source software for bioinformatics available on the web. EBImage is a package available within Bioconductor that provides general purpose functionality for image processing and analysis. Other packages required for this method are built into R. The annotated source code can be found below. Please note an item with a # in front will not run in R/R-studio, this is said to be commented out (some of the code below contains annotations in this format). The text in blue and gray can be used as input code into R/R-studio. Instead of using a fluorophore or fluorochrome *in vivo,* Chlorophyll *a* fluorescence was used as a cellular marker.

1) Set Bioconductor as source to allow R to accept inputs from the named URL directly. Load libraries constructed under Bioconductor.

   ```
   source("https://bioconductor.org/biocLite.R")
   biocLite("EBImage")
   library("EBImage")
   ```

2) Install required packages and dependencies built into R/R-studio. For first time users, install the packages first for example, in the command line of R-studio type - *Install.packages(“package name”),* and run. This will install the required package and dependencies. Note the packages only need to be installed once on your machine all subsequent times they need only be loaded as below.

   ```
   Install.packages(“tiff”)
   Install.packages(“pixmap”)
   Install.packages(“rtiff”)
   ```

3) Load packages into R/R-studio. This is required for every R session. There are two ways to do this, (1) in the R-studio user library under the packages tab check the box next to the desired package, or (2) type the code listed below to load the required libraries and their dependencies.

   ```
   library("tiff")
   library("pixmap")
   library('rtiff')
   ```

4) Create the appropriate file paths and directories for the dataset. Create file paths and directories for the pixel maps, analyzed images, and csv files created using this method. Make sure all images to be analyzed are

in one directory. Also, make sure all paths and directories match to an existing path/directory, as R/R-studio will not create new directories and will not generate an error if the path/directory listed is incorrect.

```
    # Path/directory/list for pixel map files
    map_path <- "/Users/rdel1klp/Documents/Project_Data/Hyperspectral/Growth_Curves/CellCountImages/CellCounts/Analysis/Images/"

    map_savdir <- "/Users/rdel1klp/Documents/Project_Data/Hyperspectral/Growth_Curves/CellCountImages/CellCounts/Analysis/Mapped_image"

    map_files <- list.files(map_path, pattern="tif", full.name=F)
# Path/directory/list for images for processing
    image_path <- "/Users/rdel1klp/Documents/Project_Data/Hyperspectral/Growth_Curves/CellCountImages/CellCounts/Analysis/Mapped_image/"
    image_savdir <- "/Users/rdel1klp/Documents/Project_Data/Hyperspectral/Growth_Curves/CellCountImages/CellCounts/Analysis/Image_analysis"

    image_files <- list.files(image_path, pattern="tiff", full.name=F)
# Path to save csv files from image matrix into, this is only used as a visual reference for cells/non-specific pixels
    csv_path <- "/Users/rdel1klp/Documents/Project_Data/Hyperspectral/Growth_Curves/CellCountImages/CellCounts/Analysis/Image_matrices"
```

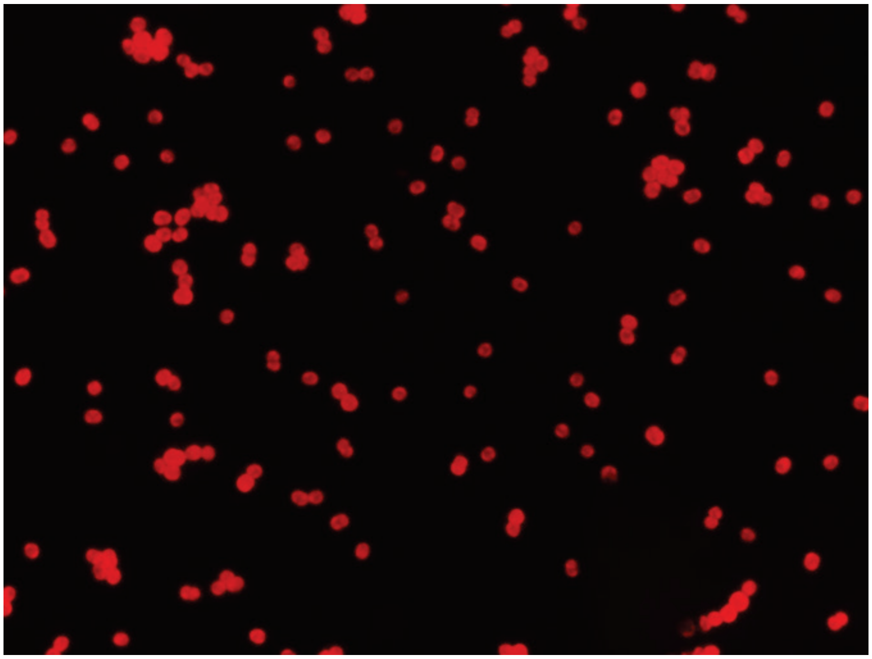
5) Create a for loop to convert images into pixel maps. A for loop is a way to repeat a sequence of instructions under certain conditions. They allow the automation of parts of code that are in need of repetition. For example, in the for loop below an RGB image is read in from a directory, converted to the red channel only, all pixels below a threshold were replaced with zero and the new image was saved in the called directory/list, then the for loop repeated this for all images in the pat. Pixel threshold can be changed to a user-defined value account for changes in the intensity of images. The purpose of this step is to reduce noise.

```
# the start of a for loop to convert tif files into pixel maps

    for (i in 1:length(map_files)){
```

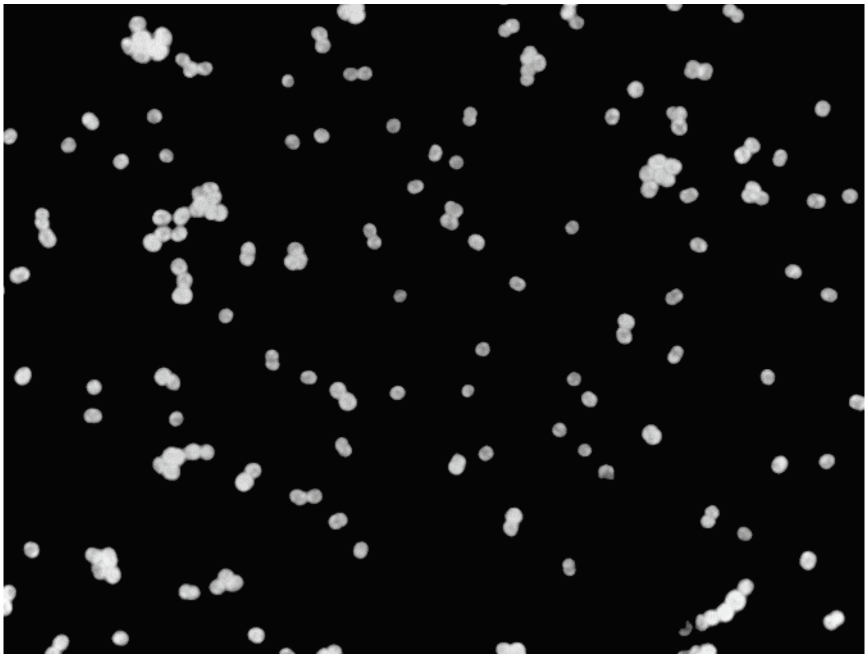
```
# name of new image file and directory to pull from
        mapped_image <- paste0(sub(".tif",replacement = "", x =
        map_files[i]),"_mapped.tiff")
        # note readTiff only accepts and outputs .tiff files
```

Figure A1. Example input image



```
# read in files from correct path
        map1 <- readTiff(paste0(map_path,map_files[i]))
# convert the red channel of the image to a pixel map/matrix.  Note this
can be changed to whatever channel particles are within.
        map2 <- as.matrix(map1@red)
# replace all pixels less than 0.3 with zero to reduce/minimize noise
        map2[map2<0.3] <- 0
# re-write the images as tiff files
        writeTiff(map2, paste0(map_savdir,"/", mapped_image))
    } # this closes the for loop
```

Figure A2. Example output image / input image for step 6.



6) Another for loop was created to process and count the new images, the code is annotated below.

# create an empty data frame to print results into. All image counts will be recorded in a data frame at the end of the for loop.

```
results <- data.frame(image = rep(NA, length(image_files)), count
= rep(NA, length(image_files)))
```

# the start of a for loop to process and count the new/edited images, the list of files was called in step 4 above.

```
for (i in 1:length(image_files)){
```

# names for the processed image and csv files, replaces '.tiff' with "_analyzed.tif" or "_matrix.csv" for all files in the path. This ensures that all files will have the same image number so they can be directly related to the parent file.

```
analyzed_image <- paste0(sub(".tiff",replacement = "", x =
image_files[i]),"_analyzed.tiff")
image_matrix <- paste0(sub(".tif",replacement = "", x =
image_files[i]),"_matrix.csv")
```

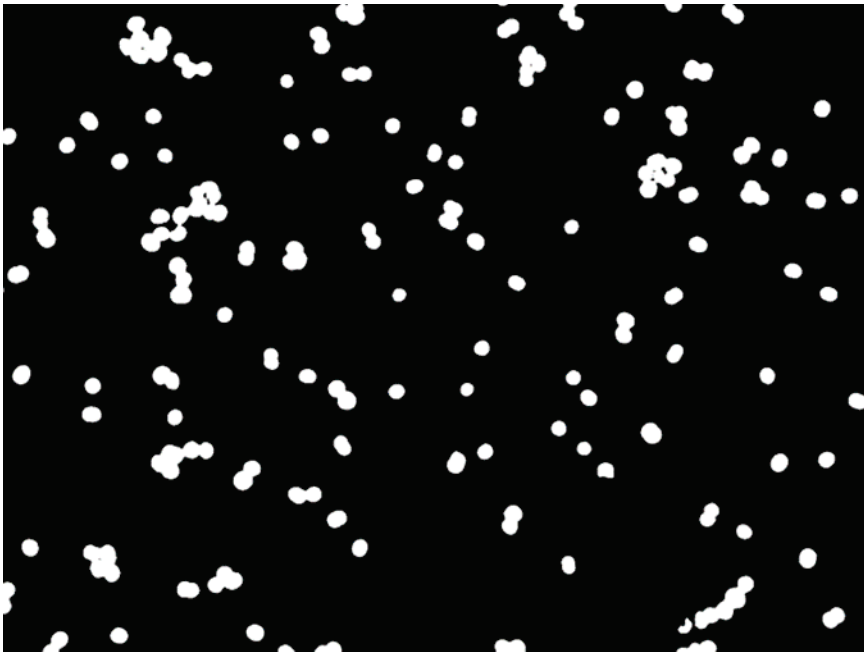
# read in all the images from image_path in the list image_files

```
img1 <- readImage(paste0(image_path, image_files[i]))
```

# apply and display a threshold to the images

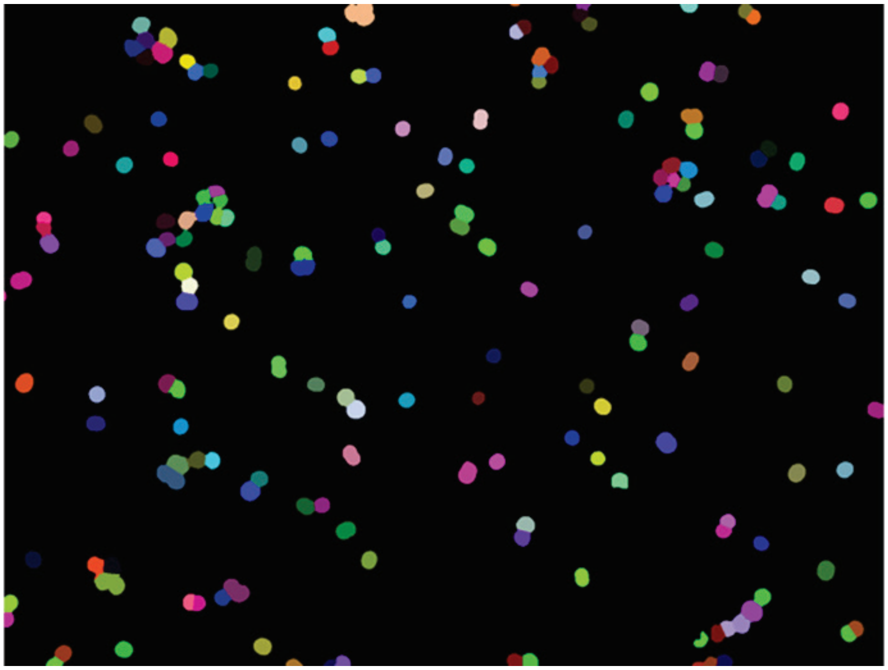
```
img2 <- thresh(img1, w=17, h=17, offset=0.001)
display(img2, method='raster') #comment out for faster runtime
```

Figure A3. Example output – note this image is not saved.



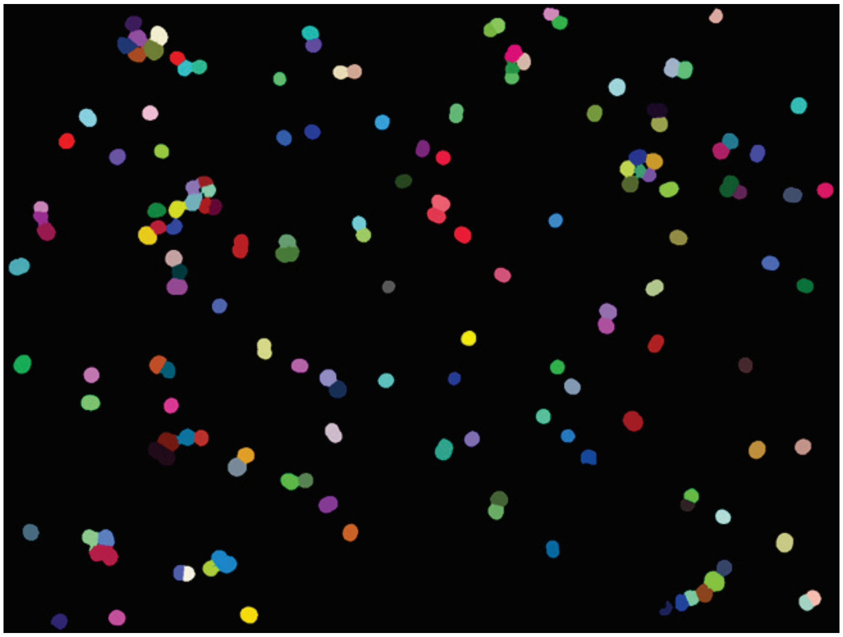
```
# apply watershed filter to images
          nmaskt <- watershed(distmap(img2), 0.5)
# fill holes within particles
           nmaskt <- fillHull(nmaskt)
# apply and display a color mask to show individual counts/particles in
different colors, this is a good visualization check for your method
          img3 <- colorLabels(nmaskt, normalize=TRUE)
          display(img3, method="raster", all=FALSE) #comment out for
#faster runtime
```

Figure A4. Example output – note this image is not saved but can be saved if no cells are to be removed on the edges, discussed below.



```
# count the maximum number of particles in each image
        count <- max(nmaskt)
# the next bit of code is used to remove cells touching the edge of the images. If it is not desired the code in green below can be omitted.
#check dimensions of output image required for border
        dims = dim(nmaskt)
# create a border for the image – this is required to remove cells touching the edge
        border = c(nmaskt[1:dims[1],1],nmaskt[1:dims[1],dims[2]],
        nmaskt[1,1:dims[2]],nmaskt[dims[1],1:dims[2]])
# used to find unique ids that touch the border/edge
        ids=unique(border[which(border !=0)])
# removes objects touching the edges
        inner = rmObjects(nmaskt, ids)
# counts the number of cells excluding those that touch the edges
        count2 <- max(inner)
# apply and display a color mask to show individual counts/particles in different colors to check that no cells are touching edges
        img4 <- colorLabels(inner, normalize=TRUE)
        display(img4, method="raster") # comment out for faster run time
```

Figure A5. Example output – note this image is saved.



# save the analyzed image

```
    writeImage(img4, files = paste0(image_savdir,"/",
    analyzed_image))
     write.csv(inner, file=paste0(csv_path,"/", image_matrix))
```

# print results in a predefined table

```
    results$image[i] <- analyzed_image
    results$count[i] <- count
    results$edge[i] <- length(ids)
    results$inner[i] <- count2
```

} # end for loop

# save all results into one table containing image name, total count, number of cells touching edges, and total count excluding cells on edges

```
    write.csv(results, paste0(image_savdir,"/colorlabels.csv"))
```

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)* August 2019	**2. REPORT TYPE** Final report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE A High-throughput Method for Counting Cyanobacteria Using Fluorescence Microscopy	**5a. CONTRACT NUMBER** W81EWF91309021 **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER** U452127
6. AUTHOR(S) Kaytee Pokrzywinski, Brandon Boyd, and Jarrell Smith	**5d. PROJECT NUMBER** 476547 **5e. TASK NUMBER** A1130 **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center Environmental Laboratory & Coastal and Hydraulics Laboratory 3909 Halls Ferry Road Vicksburg, MS 39180-6199	**8. PERFORMING ORGANIZATION REPORT NUMBER** ERDC TR-19-21
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Headquarters, U.S. Army Corps of Engineers Washington, DC 20314-1000	**10. SPONSOR/MONITOR'S ACRONYM(S)** USACE **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Harmful algal blooms (HABs) occur all over the world naturally, but are increasing in frequency largely due to eutrophication. The U.S. Army Corps of Engineers (USACE) manages over 400 freshwater waterways making cyanobacteria HABs (cyanoHABs) a significant threat to water quality. CyanoHABs have been identified as a significant problem in USACE managed waterways where districts have reported associated fish kills and waterway closures. For USACE districts, when contracted out, the return time for cyanobacterial enumeration is approximately 30 days, therefore, there is a need to provide a better approach to counting cyanobacteria. However, counting cyanobacteria is notoriously difficult due to large filaments that make conventional counting methods impractical. It is anticipated that the application of this method for routine counting of field samples will greatly reduce analysis time, and therefore, will enhance the ability to combat HABs. In addition to cyanoHAB monitoring, cell densities are required for various down-stream techniques, including nucleic acid extractions where overloading of solid phase extraction columns can cause a loss of sample. This study developed automated image processing routines that enable rapid and accurate cell counting. The results of this method compared favorably with manual counting of single-cell and filamentous cyanobacteria cultures at a fraction of the analysis time.

15. SUBJECT TERMS	Algal blooms Cyanobacterial blooms Cyanobacteria--Counting	Water quality Waterways Fluorescence microscopy

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED		41	**19b. TELEPHONE NUMBER** *(include area code)*