IMAGE PROCESSING OF DNA NANOBARCODE

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Master of Engineering

by

Manoj Nirmal Dadlani

May 2005

ABSTRACT

An image processing software package was designed to assign clearly distinguishable colors to DNA nanobarcode probes labeled with varying ratios of red and green fluorescent dyes. The DNA nanobarcode probes were produced using novel tri-strand Y-shaped DNA secondary structures. These Y-shaped DNA structures are multivalent and anisotropic, allowing for specific and controlled hybridization to other Y-shaped DNA building blocks to create a dendrimer-like DNA structure (DL-DNA). Here, the image processing of fluorescence-intensity-encoded nanobarcodes was explored using the MATLAB software environment. DNA nanobarcodes were created with varying green:red intensity ratios: 4G1R (4 green : 1 red), 2G1R, 1G1R, 1G2R and 1G4R. Additionally, these nanobarcodes incorporated specific molecular probes to target the DNA of *bacillus anthracis* (anthrax), f*rancisella tularensis* (“rabbit fever”), Ebola virus, a positive control and corona virus (SARS) respectively. Fluorescence was amplified by attaching DL-DNA to polystyrene beads through biotin-avidin interactions. The nanobarcodes were visualized using a fluorescent microscope and pseudocolor images were obtained. These images were easy to distinguish for 4G1R (bright green) and 1G4R (bright red), but were difficult to distinguish for other ratios which showed up as different shades of yellow, green and orange. Using the image processing software, these nanobarcodes were assigned highly distinguishable colors based on their green:red intensity ratio, allowing for easy and fast visual identification of the pathogens targeted. With further development, this software package will be able to assign at least 25 different and visually distinguishable colors for two and three fluorescent dyes arranged in varying ratios to create high throughput visual screening of harmful pathogens.

ACKNOWLEDGMENTS

My research project could not have been possible without the support of the entire Molecular Bioengineering Lab. I would like to especially thank Professor Dan Luo for his amazing research mentorship and for his friendship, Yougen Li and Yen Cu for providing the DNA nanobarcode images and Itay Budin who jointly worked on this project with me. In addition to those who directly participated in my research project, I would like to thank my Biological and Environmental Engineering support staff: Professor Walter, Professor Bartsch, Professor Aneshansley and Brenda Marchewka.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Au – Chemical symbol for gold

DL-DNA – Dendrimer-like DNA

xG:xR – ratio of green fluorescent dye intensity to red fluorescent dye intensity

HSV – Hue Saturation and value (colormap)

RGB – Red Green Blue (colormap)

Y-DNA – Y-shaped DNA nanostructure

INTRODUCTION

Use of nucleic acids as generic materials instead of genetic materials has sparked much research interest in recent years. Many DNA nanotechnology advances have been made by Seeman and colleagues, including the creation of various geometric objects, periodic frameworks, arrays, and scaffolds using ‘double crossover’ DNA [1]. Other, progress has also been made with linear DNA in developing systems such as biotin-avidin based gene networks, DNA conjugated Au nanocrystals, DNA-templated Au wires, and hybrid DNA-protein nanocomplexes [1, 7, 8]. However, all these advances have utilized linear DNA, restricting their effectiveness in molecular constructions [1]. To overcome this limitation, DNA building blocks of different shapes must be constructed with the ability to be incorporated into larger schemes in a controlled manner [1].

In 2004, Luo and colleagues created a new DNA building block in the shape of the letter Y. These Y-DNA building blocks are designed to have specific ‘sticky ends’, allowing them to specifically and covalently attach to other Y-DNAs in a controlled and enzyme-catalyzed means [1-2]. When the Y-DNAs are specifically attached to each other, they form a multivalent and anisotropic dendrimer-like structure, named dendrimer-like DNA (DL-DNA; Figure 1) [1-2]. There are many possible applications for DL-DNA, including a potential drug carrier, a multi-gene therapy vector, and a DNA nanobarcode [1-3].

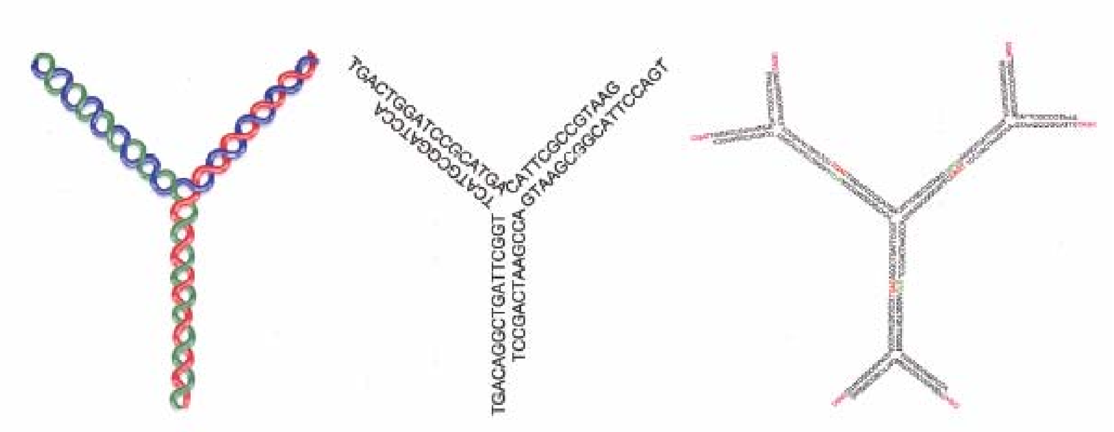

Figure 1 – (left) Schematic drawing of Y-shaped DNA. (middle) Actual single strand DNA sequences. (right) DL-DNA formed by four Y-shaped DNA building blocks [1].

In this project, the potential of using DL-DNA as a nanobarcode was explored, specifically image processing and distinguishing of the ‘code’ (green:red fluorescence intensity ratio). Fluorescence-intensity-encoded nanobarcodes were created using fluorescence-labeled Y-DNA building blocks in forming the outer layer of DL-DNA [3]. Multicolored fluorescence-intensity-encoded nanobarcodes are possible to create because both dye type and dye number can be exactly controlled (Figure 2) [3].

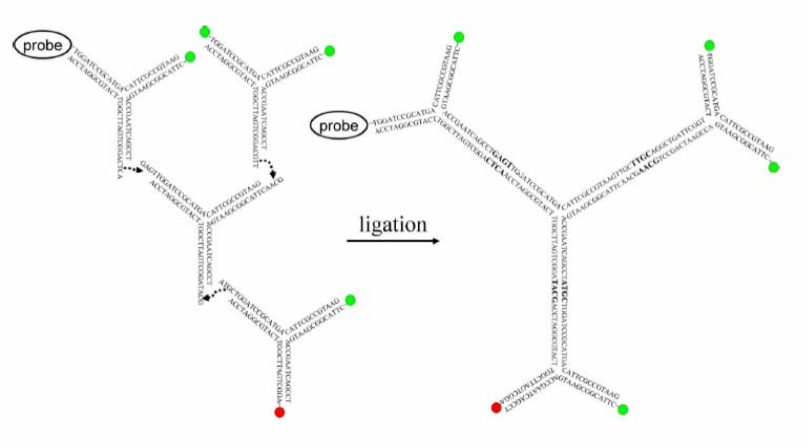

Figure 2 – Schematic drawing of DL-DNA nanobarcode. Four Y-shaped DNA molecules were covalently linked by complementary sticky end ligations. The building blocks were fluorescently labeled with green and red dyes in a controlled ratio of 4 green dye molecules to 1 red dye molecule, creating a distinct DNA nanobarcode based on fluorescence dye ratio [3].

Different ratios of different fluorescent dyes allow for the decoding of the nanobarcodes, independent of dye positioning (Figure 3) [3].

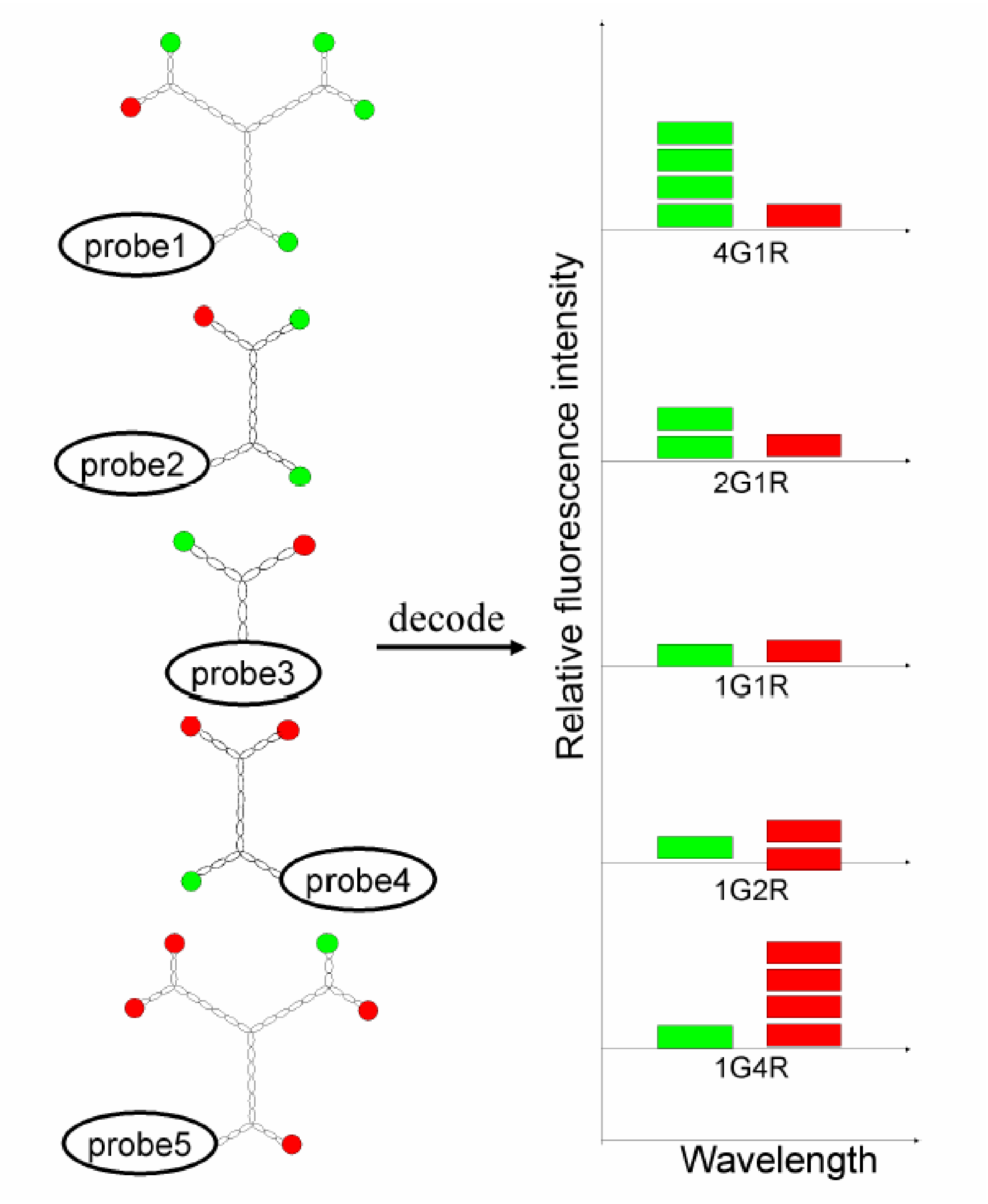

Figure 3 – Schematic drawing of DNA nanobarcode decoding based on the ratio of green:red fluorescence intensity. At each end of these DNA nanobarcodes, a molecular recognition element, a probe, was attached. This probe couples molecular sensing ability with the coding capacity of a nanobarcode [3].

Not only do these nanobarcodes have coding capability, the free reactive ends of DL-DNA allow for molecular recognition elements that can be used for molecular detection (Figure 3) [3].

Two fluorescent dyes were used to label DNA: Alexa Fluor 488 (green) and BODIPY 630/650 (red) [3]. To date, five nanobarcodes have been constructed: 4G1R, 2G1R, 1G1R, 1G2R and 1G4R, where the numbers refer to the ratio of green dye molecules to red dye molecules composing the DL-DNA structure [3]. The actual diameter of DNA nanobarcodes was less than 30nm, well below the detection limit of optical microscopy [3]. To overcome this hurdle, fluorescence signals were amplified by use of polystyrene beads in imaging and molecular detection.

Methods to detect DNA nanobarcodes that were explored in this project include fluorescence microscopy, blotting, and flow cytometry, all common and readily available laboratory procedures. However, with the vast number of ratios that can be achieved using two fluorescent dyes, it becomes extremely difficult to distinguish different red to green ratios by the naked eye. For example, 2G3R would be a very similar shade of yellow to 3G2R. By creating an image processing software that can assign distinguishable colors to different green:red ratios, DNA nanobarcodes can be visually identified by different characteristic colors specific to each ratio.

METHODS

Overview of the Software Package

Software Platforms and Environment

The software package of functions and script files was programmed using MATLAB 7.0.1 R14 in entirety and ran on a Windows XP operating system with a Pentium M 1.2 Ghz processor. Additionally, the software package can be run through the MS-DOS command prompt in Windows, thus bypassing the need for the MATLAB software environment, by installing a set of software provided on the enclosed DVD (see Appendix A for more information).

Overview

The software package was designed to give each DNA nanobarcode a visually distinct color based on the ratio of a grayscale ‘green’ channel input and a grayscale ‘red’ channel input. The software package is included with the enclosed DVD (Appendix A). Below (Figure 4) is a schematic of the algorithm for this software package:

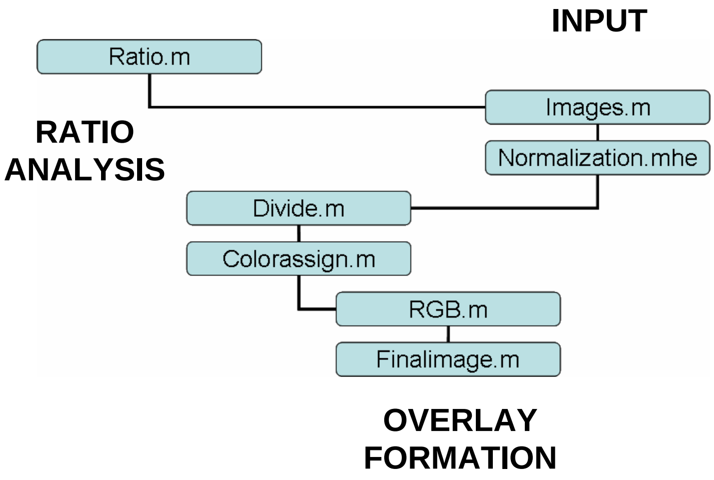

Figure 4 – Algorithm for creating visually distinct colors based on fluorescence intensity ratio of DNA nanobarcodes.

The image processing software consists of 3 parts:

1) Input Algorithm
2) Ratio Analysis Algorithm
3) Overlay Formation Algorithm

The Input algorithm prompts the user to locate the file that corresponds to the ‘green’ channel input and then prompts the user to locate the file that corresponds to the ‘red’ channel input. However, the ‘red’ dye has a stronger intensity than the ‘green’ dye and thus must be normalized. The second component of the Input algorithm is a normalization function that accounts for this difference in fluorescence intensity. Next, the Ratio algorithm takes over and calculates the ratio of green to red by dividing their grayscale intensity values (pixel by pixel). Then, using a color lookup table, specific colors are assigned to different green:red ratios. The last step, the Overlay algorithm, takes the assigned colors, creates a RGB overlay image based on the Ratio algorithm and saves the file to a specified location in the specified format.

Colormaps

Two standard colormaps were evaluated for use with this program: Red Green Blue (RGB) and Hue Saturation Value (HSV). RGB starts from black and builds up different colors by adding primary colors: red, green, and blue [4-6]. For example, if red and green is added, yellow is produced. The other colormap, HSV, is composed of Hue, what we perceive as color, Saturation is 100% for pure color and 0% for a shade of gray, and Value is related to brightness [4-6]. The advantage of using HSV over RGB is with HSV you can adjust brightness or contrast by simply changing V or S, whereas with RGB, all three color channels would need to be changed [4]. Thus, the program utilized the HSV colormap and after all the image processing occurred, converted the image to RGB as output with the built-in HSV to RGB converter in MATLAB.

Input Algorithm

The Input Algorithm is composed of two parts. First, grayscale images from the “green” and “red” channels are imported into MATLAB. Next, a normalization script runs controls of 1G1R samples to calculate a normalization factor between the actual intensities of the green and red dyes and applies the factor to the images.

Ratio Algorithm

The Ratio Algorithm divides the normalized red grayscale intensity values by the green grayscale intensity values in a pixel by pixel manner. Using these divided values, or ratios, the program consults a look up table to determine what color to assign the pixel based on the ratio value. The look up table is a standard HSV color map where S and V are set to 1.

Overlay Algorithm

The Overlay Algorithm takes the information from the ratios matched with the look up table and creates an image from it. The image is converted from HSV to RGB with MATLAB’s HSV to RGB converter and displayed, allowing the output to be saved.

Plotconverter.m

An additional software package was created to analyze flow cytometry data. Output from the flow cytometer at Cornell University's Veterinarian School is in the form of a graphical picture relating the two color intensities; however, the actual data points on the graph are not available. This package inputs the graphical data as a picture and determines values for the points on the graph in order to obtain statistical data on the relationship of the ratios. Below (Figure 5) is a schematic of the algorithm for this second software package:

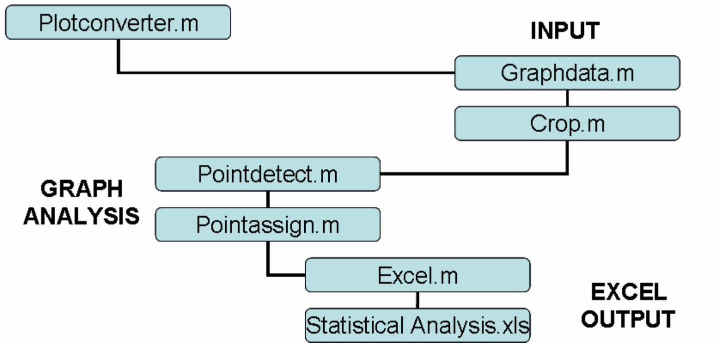

Figure 5 - Algorithm for extracting data points from flow cytometer picture graph output. These data points are used to measure green:red fluorescence intensity of DNA nanobarcode beads.

Input Algorithm

The Input Algorithm imports the graph picture from the flow cytometer output into MATLAB as a black and white image (values are either 0 for black or 1 for white). Next, the program “crops” the graph to only include the graph area without the x or y axis.

Graph Analysis Algorithm

The Graph Analysis Algorithm checks each pixel for a value of 0, corresponding to black. If the value is 0, the location of the pixel is related to the x and y axis and coordinate values are assigned and stored for that pixel.

Excel Output Algorithm

The Excel Output Algorithm takes the stored coordinate values and exports them to a Microsoft Excel file. Within Excel, statistical analysis can be performed on the data, including linear regressions and correlations.

Visualization of DNA Nanobarcode Beads

The DNA Nanobarcode was visualized in two ways: microscopy and flow cytometry. DL-DNAs with attached fluorescent dyes were no larger than 30nm, undetectable by traditional optical microscopy. In order to amplify the fluorescence signal, polystyrene microbeads (diameter = 5.5 µm) were used (Figure 6). Biotin labeled DNA capture probes bound to avidin coated polystyrene beads. Specific DNA targets were bound to the polystyrene microbeads to create a library of beads that act as “foreign particles”. Five barcodes that can detect specific “foreign particles” were created: *Bacillus anthracis* (Anthrax, 4G1R), *Francisell*a *tularensis* (“rabbit fever”, 2G1R), Ebola virus (1G1R), a positive control (1G2R), and corona virus (SARS , 1G4R). The barcodes with reporter probes bind specifically to the DNA targets on the bead and “identify” the “foreign particles”.

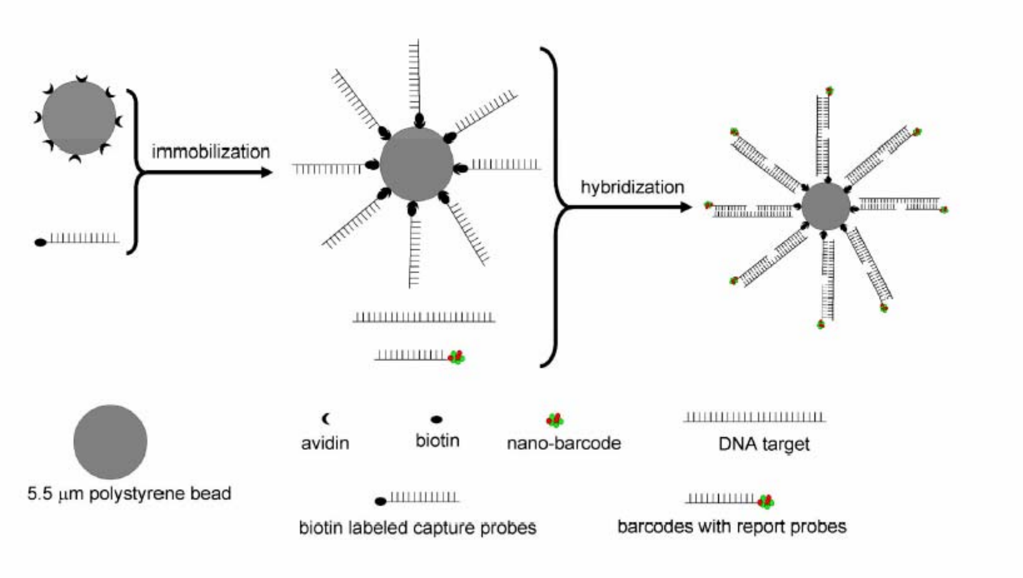

Figure 6 – Signal amplification strategy using avidin functionalized polystyrene microbeads. Biotin-labeled capture probes were attached to avidin beads and then DNA targets were captured by specific microbeads [3].

The DNA nanobarcode beads were then visualized with fluorescence microscopy and separate “green” and “red” channel images were acquired for each data set. This data was then used with Ratio.m to assign specific and distinct colors to beads based on their ratios.

Additionally, the nanobarcode beads were analyzed by flow cytometry. The data from the flow cytometer was only available as a picture graph relating the two fluorescence color intensities; the actual data points were not available. Thus, plotconverter.m was written in Matlab to extract the data point values from the picture graph.

Visualization of DNA Nanobarcode Blot

In addition to DNA nanobarcode beads, a nanobarcode blot was performed and the resultant images analyzed with the ratio software package. A DNA targets coated membrane was probed with DNA barcode labeled probes and visualized with a fluorescence microscope (**Figure 7**).

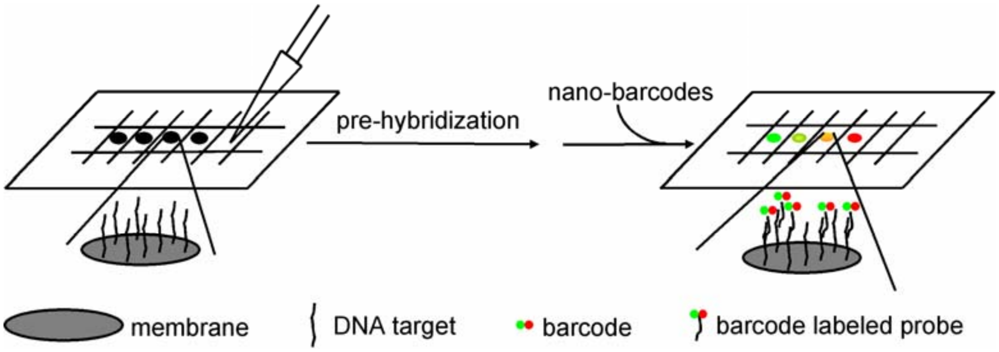

Figure 7 – Schematic drawing of multiple DNA targets with a DNA blot assay using nanobarcodes. Using DNA nanobarcodes, it is possible to detect specific DNA molecules that were manually blotted on the membrane using a fluorescence reader.

RESULTS

All results are enclosed in attached DVD (Refer to Apendix A for information on DVD contents).

DNA Nanobarcode Microbead Image Processing

Two HSV colormaps were used in microbead image processing: one for ratios 0 to 3 and one for ratios 0 to 6. Figures 8 and 10 below show the different colors associated with different ratio ranges. The first colormap, ratios 0 to 3 (Figure 8), has an increased color spread for different ratio values. As the range of ratio values increases to a value of 6 (Figure 10), the color spread of different ratio values decreases, resulting in similar colors for ratios that are closer together in value.

HSV Colormap – Ratio 0 to 3

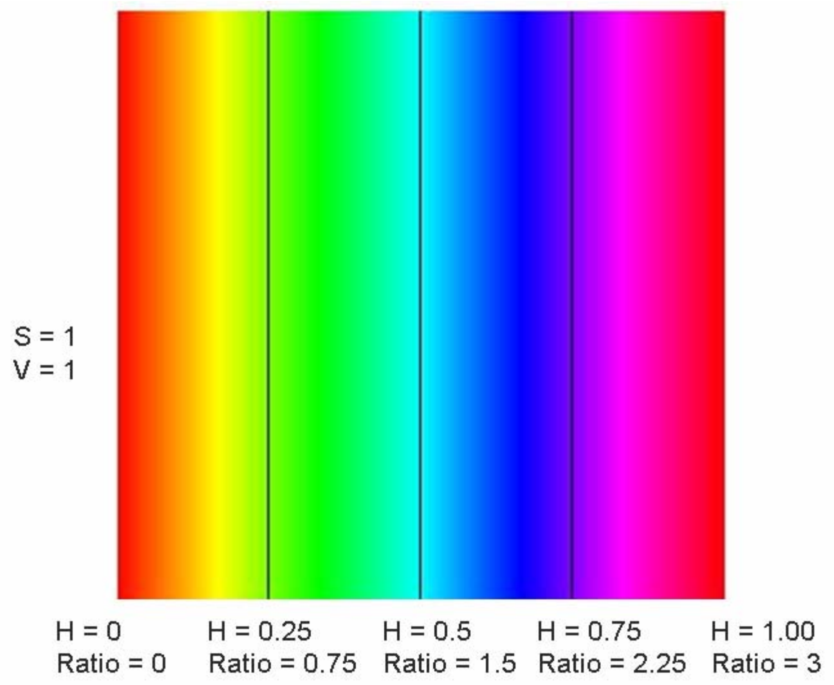

Figure 8 – HSV color map used for green: red ratio values from 0 to 3, where saturation and value are held constant at 1 and hue varies directly with ratio value.

Figure 9 below, shows the results of a three barcode sample (1G4R, 2G1R and 2G2R) that was processed using Ratio.m and compares it to a pseduocolor image obtained from the fluorescent microscope. With a three barcode sample, it is still possible to distinguish nanobarcodes from the pseudocolor image. However, Ratio.m provides a most distinguishable view of these three nanobarcodes.

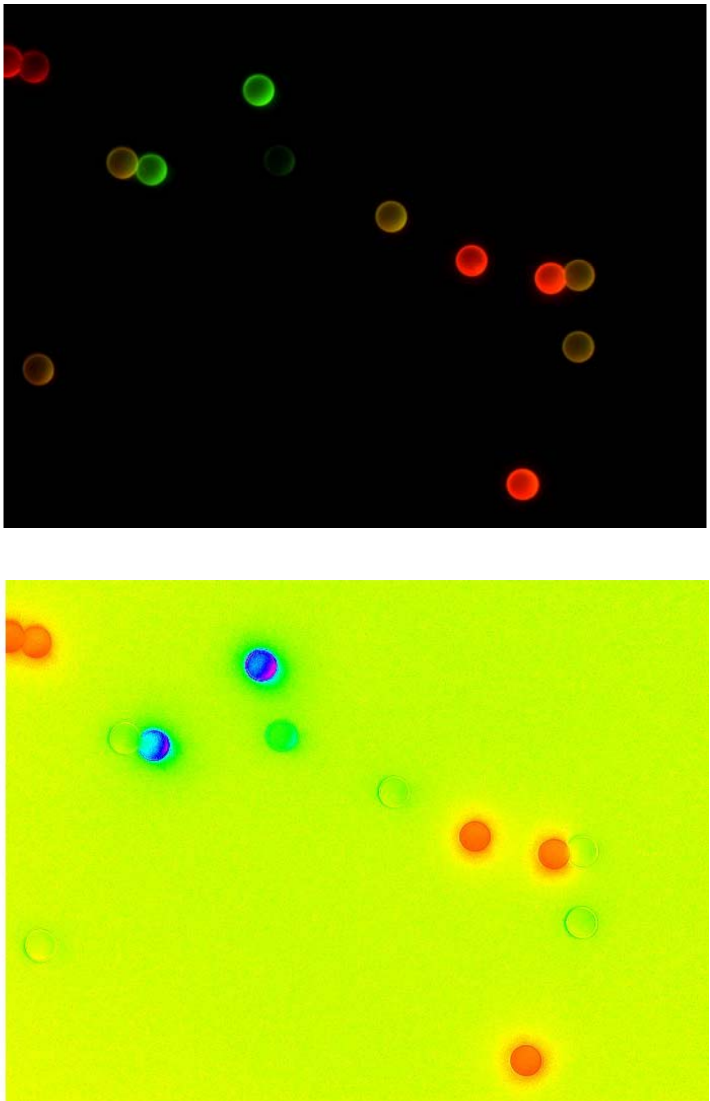

Figure 9 – (top) Pseudocolor image of three different nanobarcode microbeads, represented by the colors: orange (1G4R), yellow (1G1R) and green (2G1R). (below) Image processing results of nanobarcode beads from MATLAB simulation, represented by the colors: orange (1G4R), green (2G2R) and blue (2G1R).

HSV Colormap – Ratio 0 to 6

In order to increase the number of ratios that could be detected, the same colormap from above (Figure 8) was spread over the ratios 0 to 6, as seen below in Figure 10. This change in the ratio spread resulted in colors that were similar for smaller ratio differences. The HSV colormap below was used to detect five different nanobarcodes: 4G1R, 2G1R, 1G1R, 1G2R and 1G4R. These five nanobarcodes were used to detect DNA of five specific pathogens (one of which was actually a control): *bacillus antrhacis* (anthrax), f*rancisella tularensis* (“rabbit fever”), ebola virus, a positive control and corona virus (SARS). Figure 11 shows the base colors of these nanobarcodes and their associated DNA targets.

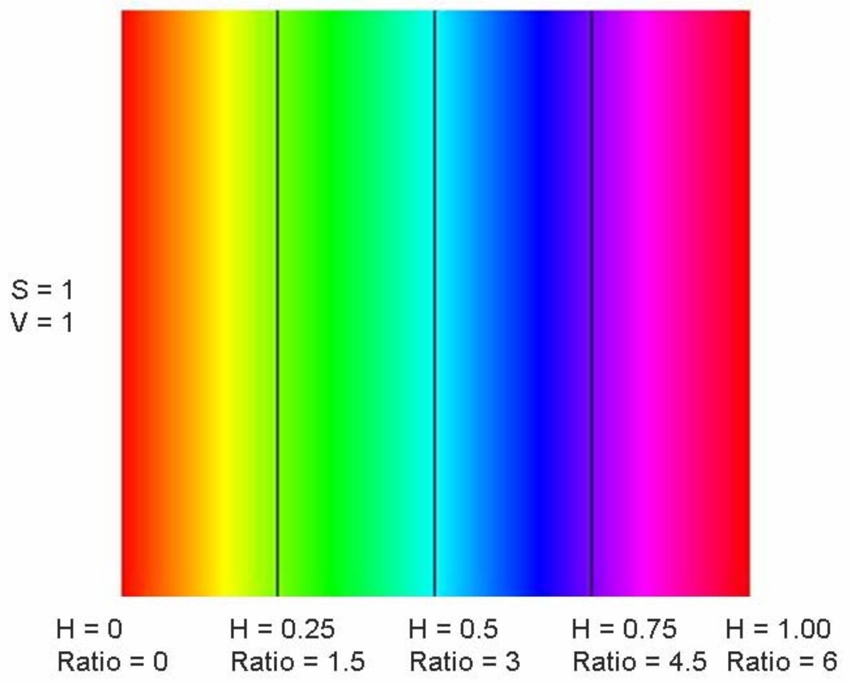

Figure 10 - HSV color map used for green:red ratio values from 0 to 6, where saturation and value are held constant at 1 and hue varies directly with ratio value.

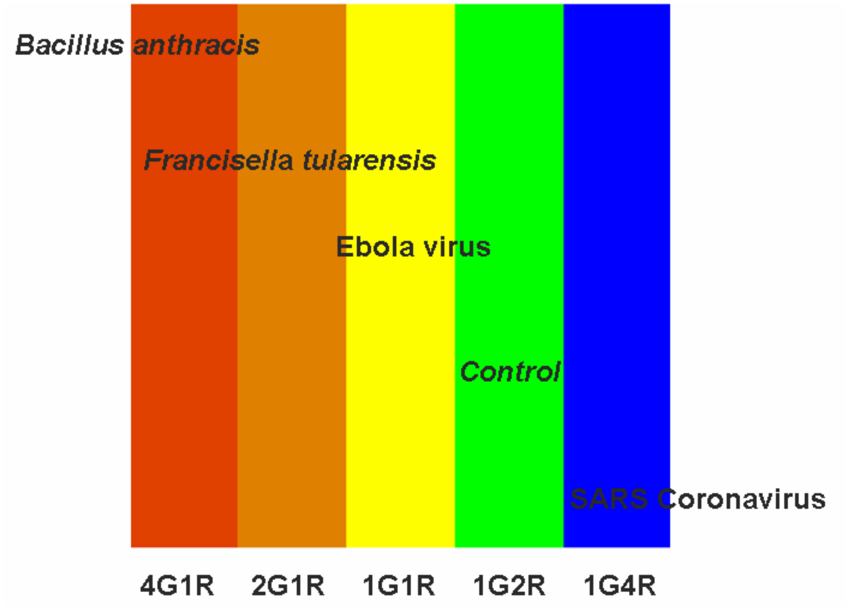

Figure 11 – Base colors for green:red ratios and the corresponding DNA targets of the DNA nanobarcode probes.

Next, Ratio.m was used on images of single nanobarcode bead samples, to create a nanobarcode lookup table based on color. Figure 12 shows the colors associated with each green:red ratio and the corresponding molecular target for both the MATLAB simulation and the pseudocolor images. With the MATLAB simulation 4G1R is red, 2G1R is orange, 1G1R is yellow, 1G2R is green and 1G4R is blue.

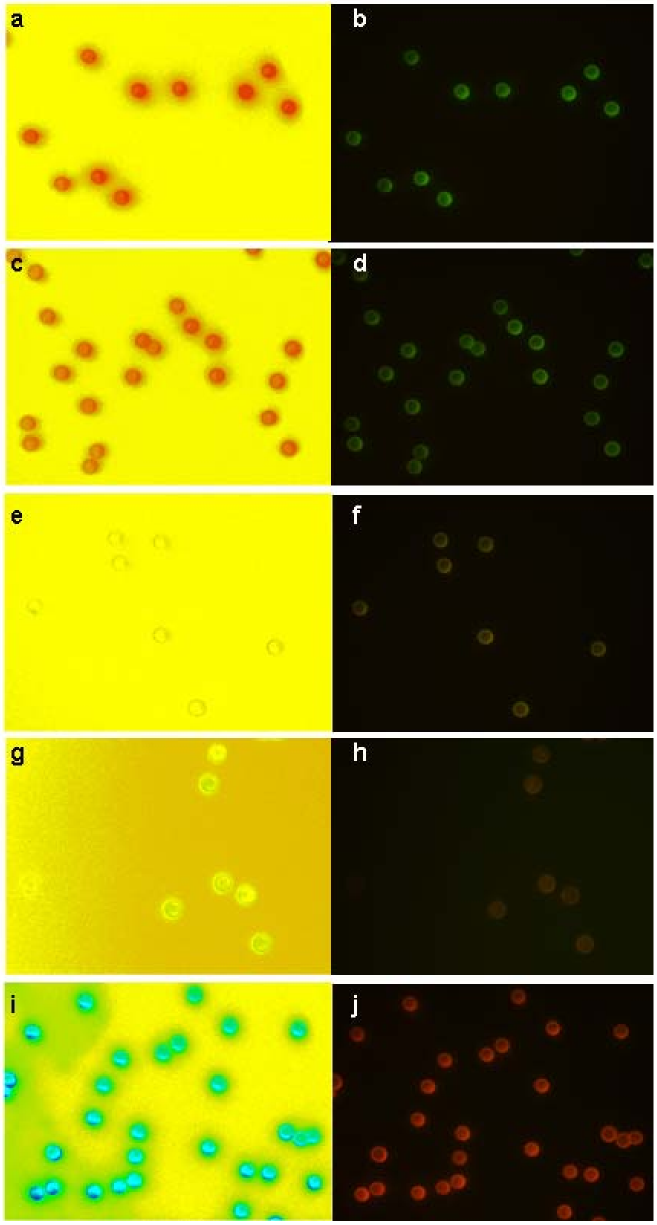

Figure 12 – Matlab simulation (a, c, e, g and i) and pseudocolor images (b,d,f,h and j) of DNA nanobarcode beads labeled with green and red dyes that correspond to particular molecular probes: (a and b) 4G1R – *bacillus anthracis* (anthrax), (c and d) 2G1R - f*rancisella tularensis* ("rabbit fever"), (e and f) 1G1R - ebola virus, (g and h) 1G2R – positive control and (i and j) 1G4R - corona virus (SARS).

While it is easy to visually distinguish 3 different DNA nanobarcodes using pseudocolor images, distinguishing 5 ratios is a much greater challenge. With Ratio.m, it is possible to assign many more than 3 ratios highly distinguishable colors based on their green:red grayscale intensity ratios. Figures 13 and 14 show the advantage of image processing for DNA nanobarcodes using Ratio.m. In Figure 13, the pseudocolor images have two color pairs that are very hard to distinguish: yellow/green and red/orange. The images processed with Ratio.m use the colors yellow/red and blue/green, which are easy to distinguish color pairs. Figure 14 also highlights the advantage of using image processing rather than simple pseudocolor images. In these images, pseudocolor produces red/orange beads, which are very similar in color, while image processing with Ratio.m outputs easier to distinguish green/blue beads.

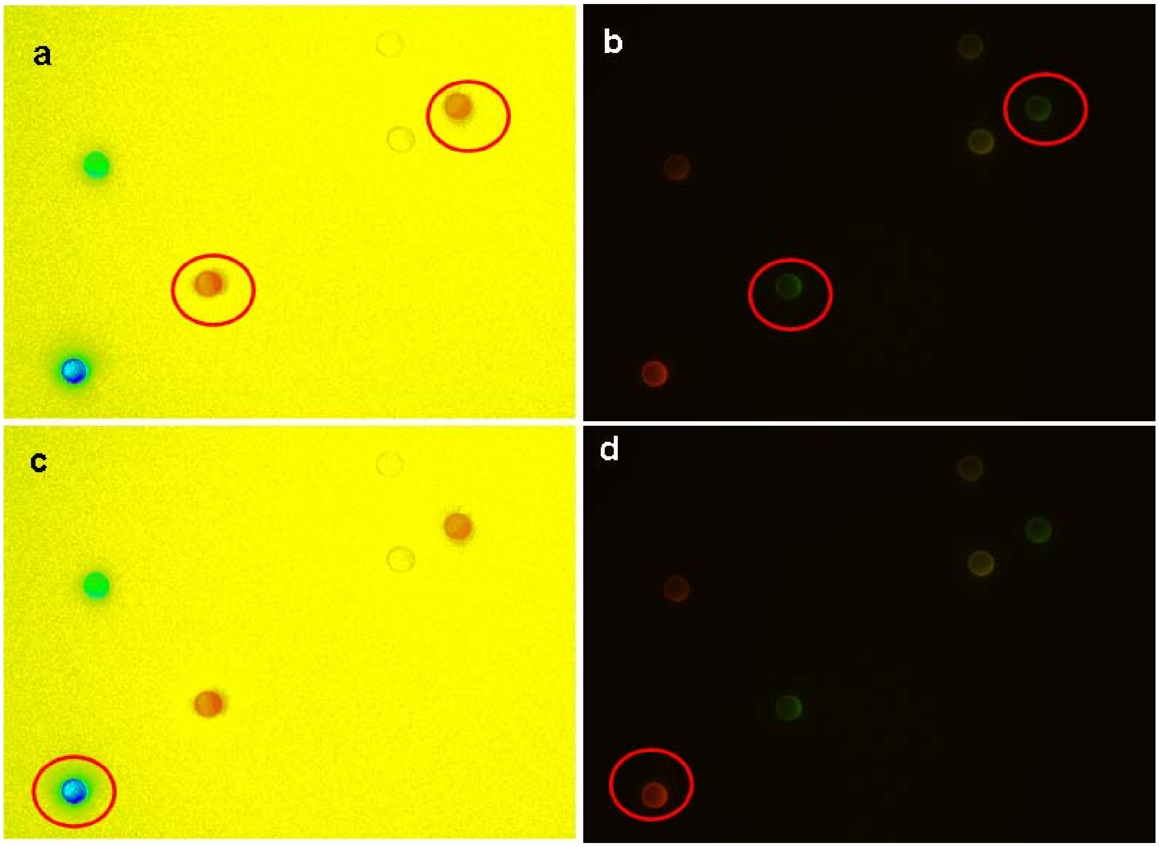

Figure 13 – The advantage of Ratio.m over simple pseudocolor images is shown with 4 different green:red ratio DNA nanobarcodes. (a) Ratio.m uses yellow/red (circled in red) beads, which are much more distinguishable than (b) the yellow/green (circled in red) beads outputted with pseudocolor. Within the same group of beads (c) Ratio.m utilizes blue (circled in red)/green beads in comparison to (d) red (circled in red)/orange beads from pseudocolor image.

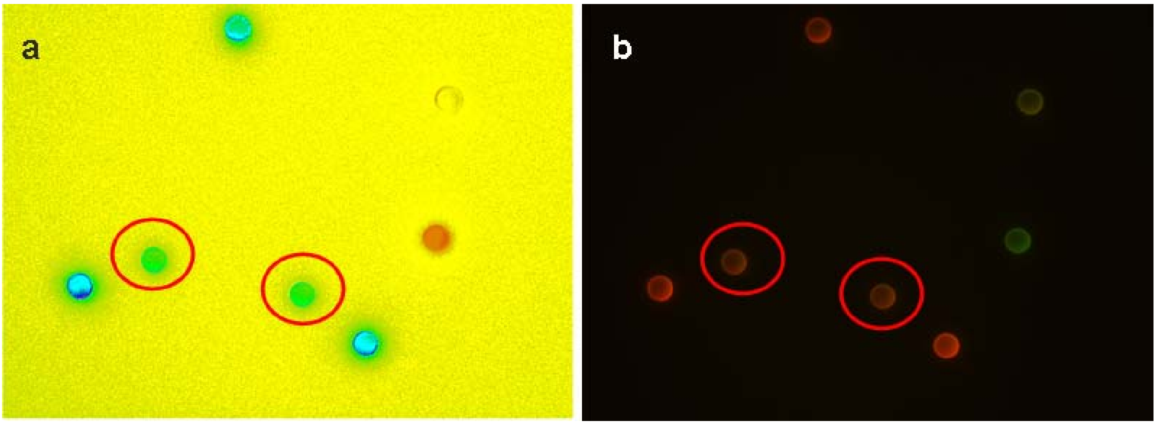

Figure 14 – This image has four distinct green:red ratio DNA nanobarcodes. (b) The pseudocolor image shows four colors: orange, red, green and yellow and the (a) Ratio.m processed image shows blue, green, orange and yellow. Comparing the two images, color distinction is easier with the Ratio.m image, especially when comparing its green (circled in red)/blue beads with the orange (circled in red)/red beads of the pseudocolor image)**.**

DNA Nanobarcode Blot Image Processing

Again, the HSV colormap with ratios from zero to six was used to evaluate the DNA nanobarcode blot using Ratio.m. Pseudocolor images obtained from fluorescence microscopy outputted four colors (Figure 15) for the four samples evaluated: green (*bacillus anthracis*), yellow (*francisella tularensis*), orange (Ebola virus) and red (SARS corona virus). Even though these four colors are distinguishable, they form two similar color pairs: green/yellow and red/orange. As more ratios are evaluated simultaneously, pseudocolor images will produces blots that are very similar in color and difficult to visually distinguish. Using Ratio.m, the following colors were produced: orange (*bacillus anthracis*), yellow (*francisella tularensis*), green (ebola virus) and blue (SARS coronavirus). These colors, as can be seen in Figure 14, are much easier to distinguish than those produced with pseudocolor.

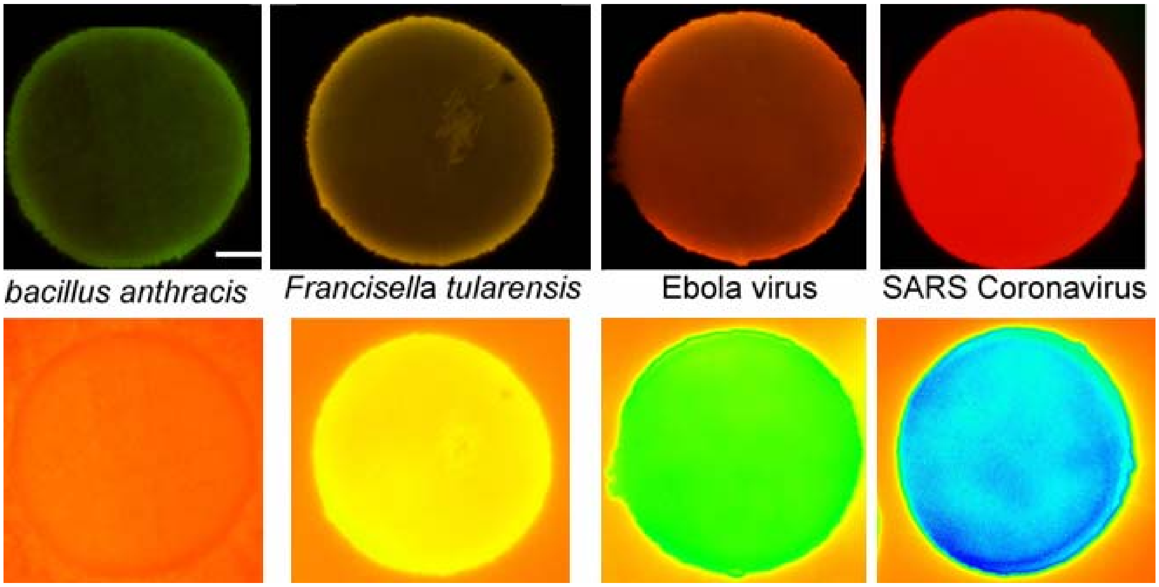

Figure 15 – (top) Pseudocolor images produced from DNA nanobarcode blots showing four colors: green (*bacillus anthracis*), yellow (*francisella tularensis*), orange *(*Ebola virus) and red (SARS corona virus). (bottom) Images outputted from Ratio.m produced more visually distinguishable colors: orange, yellow, green and blue respectively.

Plotconverter

DNA nanobarcode beads can also be evaluated using flow cytometry to measure the green:red intensity ratio. However, the flow cytometer used outputs data in graphical form, preventing the use of statistical analysis because the actual data points are unknown. Figure 16 shows samples output from the flow cytometer used in these experiments.

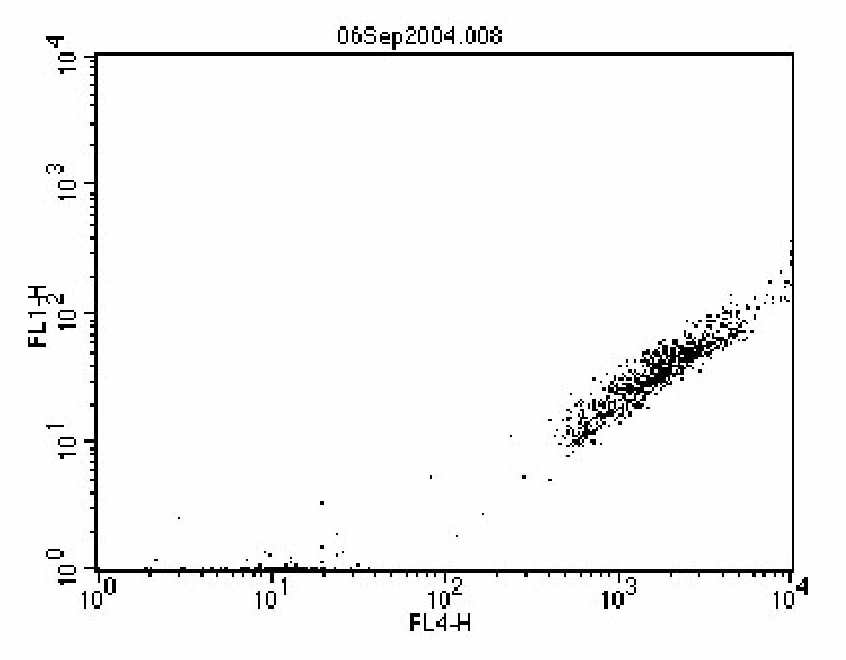

Figure 16 – Red and green intensity of DNA nanobarcode beads run through flow cytometer (>1000 beads). Data points are unknown because output is in graphical format, but will be determined using Plotconverter.m.

Plotconverter.m takes this graphical image and calculates the value of the data points. Figure 17 shows the redrawn graph from Microsoft Excel, using the data points (Appendix D) extracted with Plotconverter.m. Additionally, with Microsoft Excel, it is possible to obtain statistical analysis on the output from Figure 16. From this image, the linear regression is: y = 0.7056x – 0.7332 and the correlation value is 0.9231 as can be seen in red on Figure 17. Using this data and that from controls, it is possible to identify which molecular targets are present based on the green:red intensity ratio.

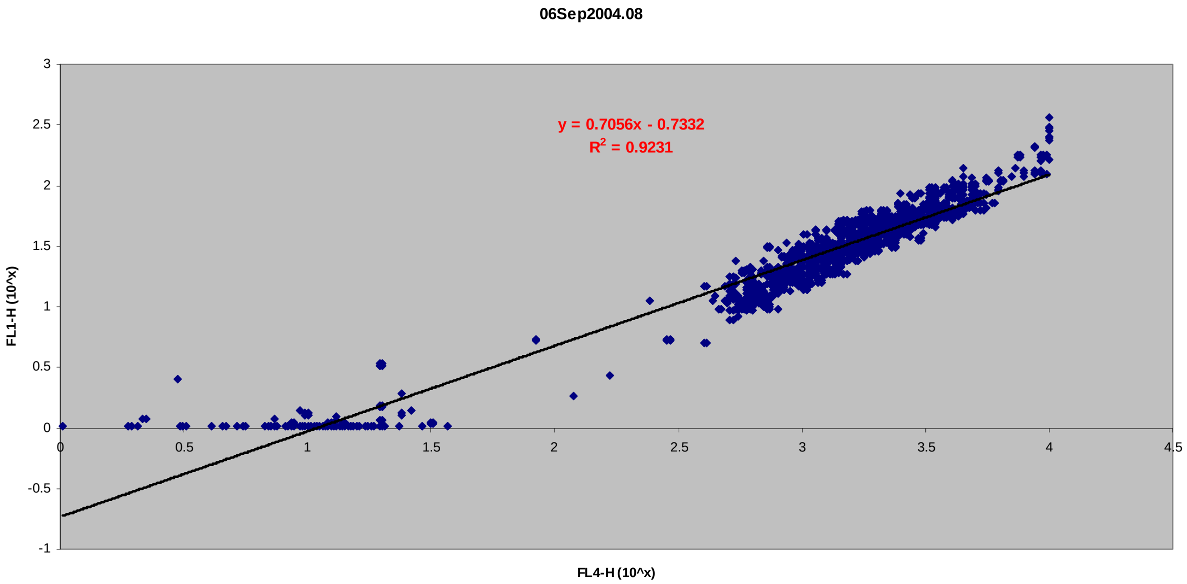

Figure 17 – Data points obtained from Plotconverter.m from Figure 16 were graphed with Microsoft Excel. Using these data points, a linear regression was performed. The equation of the line and correlation value are noted in red on the graph.

CONCLUSION AND FUTURE DIRECTION

The MATLAB image processing software package has shown to create images that are more visually distinguishable than those created using pseudocolor. So far, this software package has been used on up to five different DNA nanobarcode probes with various green:red intensity ratios. Pseudocolor images of these same five barcodes are also distinguishable, but because with so many ratios the colors are very similar, especially red/orange and green/yellow color pairs. Ratio.m overcomes these similar colors pairs by producing images with more obvious color differences. With Ratio.m, it is possible to easily visually distinguish DNA nanobarcode probes, allowing for identification of specific pathogens based on DNA hybridization.

However, the HSV colormap (Figure 10) used in the image processing thus far is not adequate for green:red ratios that are close in value. This colormap ranges from 0 to 6, but with a 0.25 difference in ratio value, the colors are very similar. Thus, a new colormap needs to be created in which different highly distinguishable colors (including intensity of those colors) are outputted for smaller ratio differences. For example, the HSV colormap in Figure 10 outputs a reddish color for a ratio value of 0.75 and an orange color for a ratio value of 1.00, both very similar colors. This new colormap should output a color such as yellow for the 0.75 ratio value and blue for the 1.00 ratio value, colors that are extremely easy to differentiate.

The first attempt at creating such a colormap is shown in Figure 18. This 25 color colormap was created with MATLAB and each color was created as a separate file. Thus, it is very simple to change the color order of the colormap. Additionally, each color is created as a gradient of light to dark from left to right. Figure 18 shows the colormap with a ratio value range of 0 to 8, another value that is easy to manipulate in MATLAB when running Ratio.m.

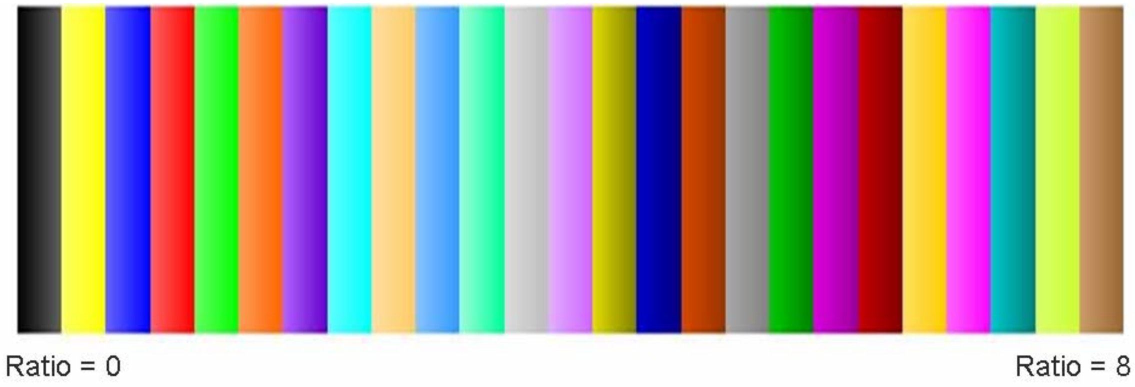

Figure 18 – Improved colormap for Ratio.m. With this colormap, colors are separated by a very small ratio difference of 0.32, allowing for ease in distinguishing DNA nanobarcodes of different green:red fluorescence intensity.

Next, Ratio.m was manipulated to utilize the new HSV colormap from Figure 18. It was found that by using a colormap with different colors separated by a very small ratio difference (0.32), more information about the DNA nanobarcode beads can be visualized and processed. Figure 19 shows the output of Ratio.m with the original HSV colormap, the new HSV colormap and the pseudocolor output. As can be seen in the output of Ratio.m with the new HSV colormap, the beads are not uniform in their green:red intensity ratio. This observation is difficult to notice with both the pseudocolor image and the image produced with Ratio.m and the original HSV colormap.

The reason for this non-uniformity is that the beads are being probed by three different structures of DNA Nanobarcodes. Figure 3 shows the various structures of the DNA nanobarcode probes: probes 1 and 5 use three y-shaped DNA building blocks, probes 2 and 4 use two y-shaped DNA building blocks and probe 3 uses one y-shaped DNA building block. When these different structures bind to the polystyrene beads, some of them align to give off a more uniform fluorescence, while others do not. Using the new HSV colormap, it is possible to assign highly distinguishable colors to DNA nanobarcode beads with similar green:red intensity ratios and because of the increased differentiation in colors it is also possible to gather information on the uniformity of the fluorescence probe.

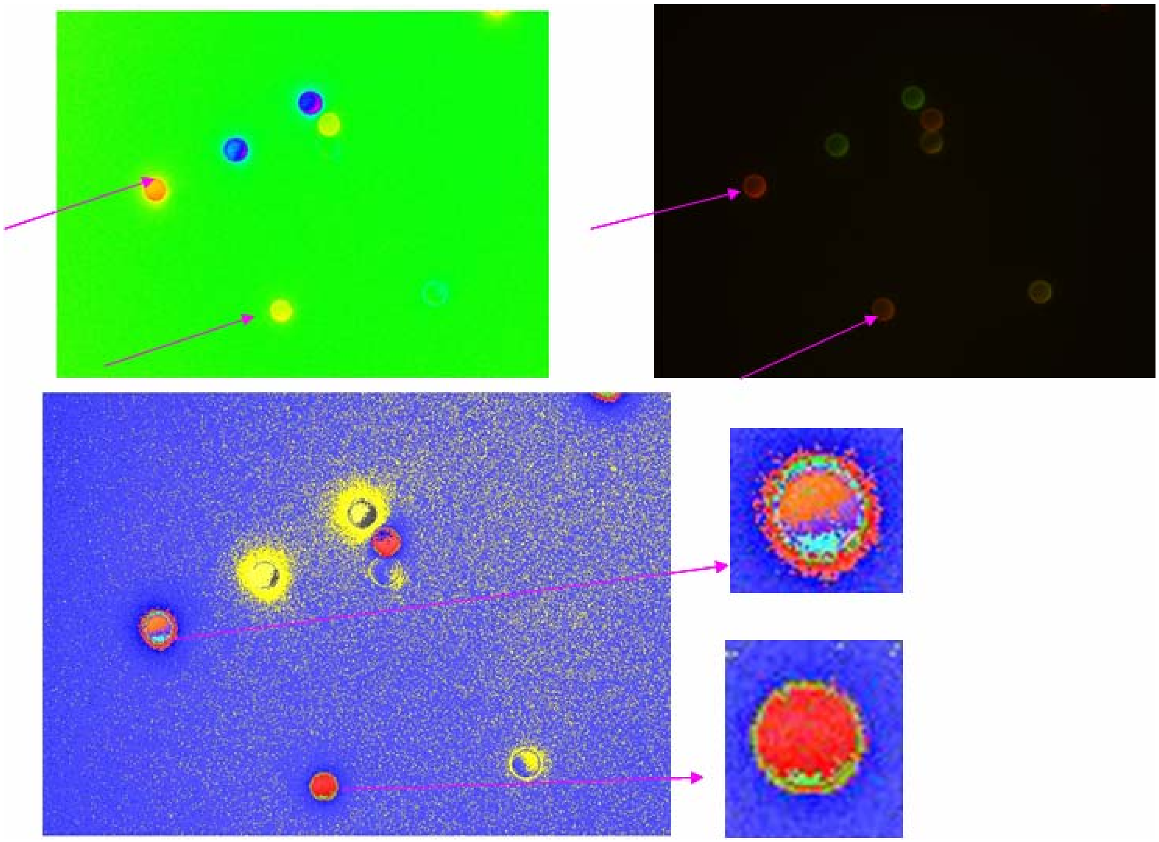

Figure 19 – (top left) Ratio.m image using original HSV colormap. (top right) Pseudocolor image produced by fluorescence microscope software. (bottom left) Ratio.m image using new HSV colormap in which three clearly identifiable ratios are observed. The blown up images on the right show that the DNA nanobarcode signals are not uniform. This information was not obtainable with pseudocolor images or with use of the original HSV colormap and Ratio.m.

Use of this new HSV colormap is the first of the advances needed to make Ratio.m a more robust program that can assign visually distinguishable colors to many more than five green:red intensity ratios. In order to improve picture quality and remove background, two other factors need to be considered when processing these images. The first is bleed through, where images taken in the green channel are actually taking data that should only be taken in the red channel and vice verse. To solve this problem, and image of solid green beads should be taken with the red fluorescence filter and one of solid red beads with the green fluorescence filter. These two images can then be used to normalize for bleed through. The second factor is to normalize for background by imaging blank beads with both the red and green filters and again using these images to normalize for fluorescence do to background.

An even greater change to the image processing done by Ratio.m is to incorporate a third dimension of the HSV colormap. Currently, saturation and value are held constant with a value of one. However, this method poses a problem because intensity is not taken into account. Thus if the ratio of green:red is 1 this could mean that the beads could be 10R:10G or 1000R:1000G. If the ratio is 10R:10G, it is more likely due to background, whereas if the ratio is 1000R:1000G, it is viable sample data. To overcome this lack of information, saturation will be used to account for red light intensity and value will be used to account for green light intensity, creating a three-dimensional color map instead of a single dimension map.

Finally, the advantage of using Y-shaped DNA building blocks to build the DNA nanobarcode is that it can theoretically have an infinite number of ‘codes’. With so many arms, it is possible to use more than two colors in the design of the DNA nanobarcode. Thus, another future goal is to manipulate Ratio.m to incorporate a third color, increasing the number of codes that are possible with the same DNA structures. With a third color and the new HSV colormap, hue will relate to the ratio value, saturation will incorporate the first color intensity, value will incorporate the third color intensity and a new visualization method, such as black dots or lines, will be used to incorporate the second color intensity. Overall, with three colors, the software will have to look at the ratio of the first two colors and if the third color is within a set percentage of the ratio, a color from the new HSV colormap will be assigned. With three colors, the coding capacity of the DNA nanobarcode will greatly increase and Ratio.m will provide a means for these codes to be decoded visually. As a result, the high processing advantages of using DNA nanobarcodes to quickly identify pathogens and other specific DNA targets can be realized visually using this software package.

APPENDIX A

Contents of enclosed DVD

Data and Results – folder contains original grayscale data for red and green channels and output from Ratio.m

MEng Thesis – Microsoft Word and Adobe PDF versions of this thesis

Misc – folder contains miscellaneous code used for testing, images and software applets that relate RGB and HSV colormaps

Plotconverter – folder contains software to run Plotconverter.m through MS-DOS, Plotconverter.m code, sample images and sample output

Ratio.m – folder contains old and new versions of Ratio.m code

References – folder contains reference documents

MEng Final Presentation – PowerPoint file used in Master of Thesis defense

APPENDIX B

MATLAB code for Ratio.m and Plotconverter.m

```
%RATIO.m

%This program reads in two grayscale images. One from the green
%fluorescent channel and one from the red fluorescence channel. It then
%divides the intensity values of the two images to create a matrix called
%'divide'. The 'divide' matrix is used to assign colors based on the ratios of the
%two images. The program uses a lookup table called colors to assign color
%values pixel by pixel to a new pseudocolor image based on the green:red
%fluorescence intensity values.


%INPUT ('images.m')

%read in first tiff file from red fluorescence channel
[imagered] = imread(input('image from RED channel\n','s'),1);

%read in second tiff file from green fluorescence channel
[imagegreen] = imread(input('image from GREEN channel\n','s'),1);


%RATIO ANALYSIS ('normalization.m', 'divide.m')

%calculate fluorescence (grayscale) ratio between images
maxdimred = size(imagered);
%gives number or rows and coloumns or red image
maxrowred = maxdimred(1); %stores the value for number of rows
maxcolumnred = maxdimred(2); %stores the value for number of columns


divide = zeros(maxrowred,maxcolumnred);
%creates matrix to store ratio values

%loop that calculates green:red fluorescence ratio pixel by pixel from inputted
%images
for i = 1:maxrowred
   for j = 1:maxcolumnred
      if imagered(i,j) ~= 0 %check for positive fluorescence value for red image
         if imagegreen(i,j) ~=0
         %check for positive fluorescence value for green image
            divide(i,j) = double(imagered(i,j))./double(imagegreen(i,j));
            %calculates green:red fluorescence intensity ratio
         end
      end
   end
end
divide = divide ./ 1.2;
%arbitrary factor for brightness of red dye,to be replaced by normalization
%function


%OVERLAY FORMATION ('colorassign.m', 'RGB.m', and 'finalimage.m')

pseudorgb = zeros(maxrowred,maxcolumnred,3);
%Empty matrix to hold pseudcolor image in 3D. The three dimensions are:
%hue, saturation and value (HSV color map)

pseudorgb(:,:,:) = 1; %assigning every pixel a white value
maxratio = max(max(divide)) %calculates the maximum of all the ratio values
meanratio = mean(mean(divide)) %calculates the mean of all the ratio values


%loops to convert gray scale image to pseudocolor
for i = 1:maxrowred
   for j = 1:maxcolumnred
      if divide(i,j) ~= 0 %check to make
      columnvalue = floor(divide(i,j)/8 * length(colors));
      %using new colors matrix (already in matlab memory) for assigning
        %colors based on ratio value
      pseudorgb(i,j,1) = colors(1,columnvalue,1);
        %assign hue value to pseudocolor image
      pseudorgb(i,j,2) = colors(1,columnvalue,2);
        %assign saturation value to pseudocolor image
      pseudorgb(i,j,3) = colors(1,columnvalue,3);
        %assign value value to pseudocolor image
      end
   end
end

imshow(pseudorgb); %display newly created pseudocolor image
```

```
%code for using the old hsv colormap (ratio 0 to 6)
%loops to convert gray scale image to pseudocolor
%for i = 1:maxrowred
%    for j = 1:maxcolumnred
%        if divide(i,j) ~= 0
%        pseudohsv(i,j,1) = divide(i,j)/6;
%        end
%    end
%end
%
%pseudorgb = hsv2rgb(pseudohsv);
%imshow(pseudorgb);
```

```
function plotconverter (number)

%PLOTCONVERTER.m
%This function takes in a a log plot generated by a flow spectrometer and
%creates an excel file containing the data in exponential form.
%ie the data point 2.0 is actually 10^2.0=100. This is done for data
%analysis purposes in excel.

%Written by Itay Budin and Manoj Dadlani

[plotmap,map] = imread(input('plot filename?\n','s'),1);

%The dimensions of matrix
[maxrow, maxcol] = size(plotmap);

% Finding the coordinates of the two "origins"

counter = 0; %counter of number of black pixels
maxcount = 0; %the highest number of consecutive black pixels in the matrix
xhighest = 0; %the x-coordinate of the bottom left origin
yhighest = 0; %the y-coordinate of the bottom left origin

%Loop through the matrix to find the most consecutive black pixels, ie the
%bottom-left origin

%x-coordinate of origin
for i=1:1:(maxcol/2)
    xcounter = 0;
    for j=1:1:maxrow

         if plotmap(j,i) < 200
             counter= counter + 1;
         else
             counter = 0;
         end
         if counter > maxcount
             xhighest = [i];
             maxcount = counter;
         end
     end

end

%creates x boundary taking into account 2 pixel thick axis from graph
xmin = xhighest+2;
```

```
%y-coordinate of origin
counter = 0;
yhighest = 0;
maxcount = 0;

for j=maxrow:-1:(maxrow/2)
    ycounter = 0;
    for i=1:1:maxcol
        if plotmap (j,i) < 200
            counter= counter + 1;
        else
            counter = 0;
        end
        if counter > maxcount
            yhighest = [j];
            maxcount = counter;
        end
    end
end

%creates y boundary taking into account 2 pixel thick axis from graph
ymax = yhighest -2;


graphorigin = [xmin,ymax]; %actual bottom left origin of flow cytometer graph

counter = 0; %counter of number of black pixels
maxcount = 0; %the highest number of consecutive black pixels in the matrix
xhighest = 0; %the x-coordinate of this origin
yhighest = 0; %the y-coordinate of this origin

%Loop through the matrix to find the most consecutive black pixels, ie the
%top right origin
  for i = maxcol:-1:(maxcol/2)
    for j = maxrow:-1:1
         if plotmap(j,i) < 200
             counter= counter + 1;
         else
             counter = 0;
         end
         if counter >= maxcount
             xhighest = [i];
             yhighest = [j];
             maxcount = counter;
         end
     end
     counter = 0;
  end

%creates upper x-boundary
xmax = xhighest -1;

%creates "upper" y-boundary, referred to as ymin because plotmap starts at
%top left of image
ymin = yhighest + 2;

upperbound = [xmax,ymin];

cropplot = plotmap(ymin:ymax, xmin:xmax);

%We assign two variables for the cropped plot dimensions
[rowdim coldim] = size(cropplot);

%Going through the cropped plot to find the black pixels:
%Note: a more sophisticated algorithm for determing the location of
%a data point can be added here.

%datapts is an n x 2 matrix which contains the x and y (standardized)
%coordinates of the data point. The coordinates are standardized (in
%numbering) from left to right and bottom to top. Furthermore, they are
%represented as a ratio relative to the max dimensions of the matrix. Thus,
%the coordinates range from zero to one in each direction.

datapts = [0 0];                  %initialization of datapts
for i=1:1:(coldim)
    for j=1:1:(rowdim)
        if cropplot(j,i) < 150
%if a dark (150 is set as an arbitrary value) pixel is found...
            datapts = [datapts ; (i./coldim) ((rowdim+1-j)./rowdim)];
                                    %...its coordinate is added to datapts
        end
    end
end

%The first entry, (0,0), of datapts is removed
datasize = size(datapts);
datapts = datapts(2:datasize(1),:);
datasize = size(datapts);
```

```
%adjust the datapoints so that xmax of the graph corresponds to a value of
%10,000 and ymax of the graph corresponds to a value of 10,000
datapts = datapts.*4;

%Convert points from log scale to non-log scale
pnts = datasize(1) * datasize(2);
%for i = 1:pnts
%    datapts(i) = 10.^datapts(i);
%end

%The matrix containing the relative coordinates of the black pixels is exported
%to a Microsoft Excel file for further analysis (the directory for file export
%should be set here)
savelocation = input('save location?\n','s');
wk1write(savelocation,datapts);
```

APPENDIX C

MATLAB code of color files for new HSV color map

%COLORS.m

```
%black in RGB
%start with RGB = 0,0,0 (black)
%end with   RGB = 100,100,100 (dark grey)

black = zeros(300,100,3);%
dimensions = size(black);%

row = dimensions(1);
col = dimensions(2);

rstart = 0/255;%
rend = 100/255;%

gstart = 0/255;%
gend = 100/255;%

bstart = 0/255;%
bend = 100/255;%

rinc = (rend - rstart)/col;
ginc = (gend - gstart)/col;
binc = (bend - bstart)/col;

for i = 1:col
   black(:,i,1) = rstart;%
   rstart = rstart + rinc;

   black(:,i,2) = gstart;%
   gstart = gstart + ginc;

   black(:,i,3) = bstart;%
   bstart = bstart + binc;
end
```

```
%blue in RGB
%start with RGB = 100,100,255 (light blue)
%end with   RGB = 0,0,255 (blue)

%brown in RGB
%start with RGB = 206,157,108 (light brown)
%end with   RGB = 153,102,51 (brown)

%cyan in RGB
%start with RGB = 100,255,255 (light cyan)
%end with   RGB = 0,255,255 (cyan)

%darkbrown in RGB
%start with RGB = 218,73,0 (light darkbrown)
%end with   RGB = 153,51,0 (darkbrown)

%darkcyan in RGB
%start with RGB = 0,200,200 (light darkcyan)
%end with   RGB = 0,128,128 (darkcyan)

%darkgreen in RGB
%start with RGB = 0,200,0 (light darkgreen)
%end with   RGB = 0,128,0 (darkgreen)

%darkyellow in RGB
%start with RGB = 227,222,0 (light maroon)
%end with   RGB = 128,128,0 (maroon)

%fuchsia in RGB
%start with RGB = 255,100,255 (light fuchsia)
%end with   RGB = 255,0,255 (fuchsia)

%green in RGB
%start with RGB = 100,255,100 (light green)
%end with RGB = 0,255,0 (green)

%lavender in RGB
%start with RGB = 228,175,255 (light lavender)
%end with   RGB = 204,102,255 (lavender)

%lightblue in RGB
%start with RGB = 137,196,255 (light lightblue)
%end with   RGB = 51,153,255 (lightblue)
```

%**lightgray** in RGB
%start with RGB = 220,220,220 (light lightgray)
%end with   RGB = 190,190,190 (lightgray)

%**limegreen** in RGB
%start with RGB = 217,255,109 (light limegreen)
%end with   RGB = 204,255,51 (limegreen)

%**marigold** in RGB
%start with RGB = 255,226,101 (light marigold)
%end with   RGB = 255,204,0 (marigold)

%**maroon** in RGB
%start with RGB = 200,0,0 (light maroon)
%end with   RGB = 128,0,0 (maroon)

%**medgray** in RGB
%start with RGB = 140,140,140 (light medgray)
%end with   RGB = 105,105,105 (medgray)

%**navy** in RGB
%start with RGB = 0,0,200 (light navy)
%end with   RGB = 0,0,128 (navy)

%**orange** in RGB
%start with RGB = 255,155,87 (light orange)
%end with   RGB = 255,102,0 (orange)

%**peach** in RGB
%start with RGB = 255,224,163 (light peach)
%end with   RGB = 255,204,102 (peach)

%**purple** in RGB
%start with RGB = 178,101,255 (light purple)
%end with   RGB = 102,0,204 (purple)

%**red** in RGB
%start with RGB = 255,100,100 (light red)
%end with   RGB = 255,0,0 (red)

%**seagreen** in RGB
%start with RGB = 155,255,215 (light seagreen)
%end with   RGB = 0,255,153 (seagreen)

```
%violet in RGB
%start with RGB = 222,0,222 (light violet)
%end with   RGB = 153,0,153 (violet)

%yellow in RGB
%start with RGB = 255,255,100 (light yellow)
%end with   RGB = 255,255,0 (yellow)


%colors creates MOLBEL color map

colors = [black yellow blue red green lightgray purple cyan peach lightblue
seagreen orange lavender darkyellow navy darkbrown medgray darkgreen
violet marigold maroon fuchsia darkcyan limegreen brown];

imshow(colors);
```

APPENDIX D

Data points extracted using Plotconverter.m

x-value	y-value	x-value	y-value	x-value	y-value
0.011527	0.015625	2.916427	1.312500	2.708934	1.187500
0.276657	0.015625	2.916427	1.296875	2.708934	1.171875
0.288184	0.015625	2.916427	1.281250	2.708934	1.125000
0.311239	0.015625	2.916427	1.265625	2.708934	1.093750
0.334294	0.078125	2.916427	1.218750	2.708934	1.078125
0.345821	0.078125	2.916427	1.203125	2.708934	1.062500
0.472622	0.406250	2.916427	1.187500	2.708934	0.968750
0.484150	0.015625	2.916427	1.156250	2.708934	0.890625
0.495677	0.015625	2.916427	1.140625	2.720461	1.250000
0.507205	0.015625	2.927954	1.421875	2.720461	1.187500
0.610951	0.015625	2.927954	1.406250	2.720461	1.171875
0.657061	0.015625	2.927954	1.312500	2.720461	1.125000
0.668588	0.015625	2.927954	1.296875	2.720461	1.093750
0.714697	0.015625	2.927954	1.281250	2.720461	1.078125
0.737752	0.015625	2.927954	1.265625	2.720461	1.062500
0.749280	0.015625	2.927954	1.218750	2.720461	0.968750
0.829971	0.015625	2.927954	1.203125	2.720461	0.890625
0.841499	0.015625	2.927954	1.187500	2.731988	1.375000
0.853026	0.015625	2.927954	1.156250	2.731988	1.234375
0.864553	0.078125	2.927954	1.140625	2.731988	1.187500
0.864553	0.015625	2.939481	1.531250	2.731988	1.109375
0.876081	0.015625	2.939481	1.359375	2.731988	0.984375
0.910663	0.015625	2.939481	1.343750	2.743516	1.093750
0.922190	0.015625	2.939481	1.328125	2.743516	1.078125
0.933718	0.046875	2.939481	1.234375	2.743516	1.062500
0.933718	0.031250	2.939481	1.218750	2.743516	1.046875
0.933718	0.015625	2.939481	1.203125	2.743516	1.015625
0.945245	0.046875	2.951009	1.437500	2.743516	1.000000
0.945245	0.031250	2.951009	1.421875	2.743516	0.968750
0.945245	0.015625	2.951009	1.406250	2.743516	0.921875
0.968300	0.140625	2.951009	1.390625	2.755043	1.296875
0.968300	0.015625	2.951009	1.359375	2.755043	1.281250
0.979827	0.015625	2.951009	1.343750	2.755043	1.046875
0.991354	0.125000	2.951009	1.312500	2.755043	1.015625
0.991354	0.109375	2.951009	1.296875	2.755043	1.000000
0.991354	0.015625	2.951009	1.281250	2.755043	0.984375
1.002882	0.125000	2.951009	1.265625	2.766571	1.296875
1.002882	0.109375	2.951009	1.250000	2.766571	1.281250
1.002882	0.015625	2.951009	1.234375	2.766571	1.046875
1.014409	0.015625	2.951009	1.218750	2.766571	1.015625
1.025937	0.015625	2.951009	1.203125	2.766571	1.000000

1.037464	0.015625	2.951009	1.125000	2.766571	0.984375
1.048991	0.015625	2.962536	1.468750	2.778098	1.312500
1.060519	0.015625	2.962536	1.453125	2.778098	1.218750
1.072046	0.015625	2.962536	1.421875	2.778098	1.203125
1.083573	0.046875	2.962536	1.406250	2.778098	1.156250
1.083573	0.031250	2.962536	1.390625	2.778098	1.140625
1.083573	0.015625	2.962536	1.312500	2.778098	1.109375
1.095101	0.046875	2.962536	1.250000	2.778098	1.078125
1.095101	0.031250	2.962536	1.234375	2.778098	1.062500
1.095101	0.015625	2.962536	1.218750	2.778098	1.031250
1.106628	0.015625	2.962536	1.203125	2.778098	1.015625
1.118156	0.093750	2.974063	1.468750	2.778098	1.000000
1.118156	0.062500	2.974063	1.453125	2.778098	0.984375
1.118156	0.015625	2.974063	1.421875	2.778098	0.968750
1.129683	0.015625	2.974063	1.406250	2.789625	1.328125
1.141210	0.046875	2.974063	1.390625	2.789625	1.296875
1.141210	0.031250	2.974063	1.312500	2.789625	1.281250
1.141210	0.015625	2.789625	1.250000	2.858790	1.484375
1.152738	0.046875	2.789625	1.187500	2.858790	1.328125
1.152738	0.031250	2.789625	1.171875	2.858790	1.312500
1.152738	0.015625	2.789625	1.156250	2.858790	1.296875
1.164265	0.015625	2.789625	1.140625	2.858790	1.281250
1.175793	0.015625	2.789625	1.109375	2.858790	1.265625
1.187320	0.015625	2.789625	1.046875	2.858790	1.250000
1.198847	0.015625	2.789625	1.031250	2.858790	1.187500
1.210375	0.015625	2.789625	0.984375	2.858790	1.093750
1.233429	0.015625	2.801153	1.312500	2.858790	1.078125
1.244957	0.015625	2.801153	1.250000	2.858790	1.062500
1.256484	0.015625	2.801153	1.234375	2.858790	1.015625
1.268012	0.015625	2.801153	1.218750	2.858790	1.000000
1.291066	0.531250	2.801153	1.203125	2.858790	0.984375
1.291066	0.515625	2.801153	1.187500	2.870317	1.500000
1.291066	0.187500	2.801153	1.125000	2.870317	1.484375
1.291066	0.171875	2.801153	1.078125	2.870317	1.328125
1.291066	0.062500	2.801153	1.062500	2.870317	1.312500
1.291066	0.015625	2.801153	1.046875	2.870317	1.296875
1.302594	0.531250	2.801153	1.015625	2.870317	1.281250
1.302594	0.515625	2.801153	1.000000	2.870317	1.265625
1.302594	0.187500	2.801153	0.968750	2.870317	1.250000
1.302594	0.171875	2.812680	1.187500	2.870317	1.187500
1.302594	0.062500	2.812680	1.156250	2.870317	1.093750
1.302594	0.015625	2.812680	1.140625	2.870317	1.078125
1.314121	0.015625	2.812680	1.125000	2.870317	1.062500
1.371758	0.015625	2.812680	1.109375	2.870317	1.015625
1.383285	0.281250	2.812680	1.093750	2.870317	1.000000
1.383285	0.125000	2.812680	1.078125	2.870317	0.984375
1.383285	0.109375	2.812680	1.062500	2.881844	1.250000
1.417867	0.140625	2.812680	1.046875	2.881844	1.218750
1.463977	0.015625	2.812680	1.031250	2.881844	1.203125

1.498559	0.046875	2.824207	1.187500	2.881844	1.187500
1.498559	0.031250	2.824207	1.156250	2.881844	1.156250
1.510086	0.046875	2.824207	1.140625	2.881844	1.140625
1.510086	0.031250	2.824207	1.125000	2.881844	1.109375
1.567723	0.015625	2.824207	1.109375	2.881844	1.093750
1.925072	0.734375	2.824207	1.093750	2.893372	1.312500
1.925072	0.718750	2.824207	1.078125	2.893372	1.296875
2.074928	0.265625	2.824207	1.062500	2.893372	1.281250
2.224784	0.437500	2.824207	1.046875	2.893372	1.234375
2.386167	1.046875	2.824207	1.031250	2.893372	1.218750
2.455331	0.734375	2.835735	1.296875	2.893372	1.203125
2.455331	0.718750	2.835735	1.281250	2.893372	1.125000
2.466859	0.734375	2.835735	1.265625	2.904899	1.468750
2.466859	0.718750	2.835735	1.156250	2.904899	1.328125
2.605187	1.171875	2.835735	1.140625	2.904899	1.296875
2.605187	0.703125	2.835735	1.093750	2.904899	1.281250
2.616715	1.171875	2.835735	1.078125	2.904899	1.234375
2.616715	0.703125	2.835735	1.062500	2.904899	1.218750
2.639769	1.046875	2.847262	1.375000	2.904899	1.203125
2.651297	1.093750	2.847262	1.171875	2.904899	1.187500
2.662824	0.984375	2.847262	1.125000	2.904899	1.171875
2.674352	0.984375	2.847262	1.078125	2.904899	1.125000
2.685879	1.171875	2.847262	1.062500	2.904899	1.109375
2.685879	1.046875	2.847262	1.015625	2.904899	0.984375
2.697406	1.031250	2.847262	1.000000	2.916427	1.421875
2.708934	1.250000	2.858790	1.500000	2.916427	1.406250
2.974063	1.250000	3.031700	1.296875	3.077810	1.312500
2.974063	1.234375	3.031700	1.281250	3.077810	1.234375
2.974063	1.218750	3.031700	1.265625	3.077810	1.218750
2.974063	1.203125	3.031700	1.250000	3.077810	1.203125
2.985591	1.515625	3.031700	1.234375	3.089337	1.531250
2.985591	1.500000	3.031700	1.218750	3.089337	1.437500
2.985591	1.484375	3.031700	1.203125	3.089337	1.421875
2.985591	1.437500	3.043228	1.515625	3.089337	1.406250
2.985591	1.390625	3.043228	1.453125	3.089337	1.390625
2.985591	1.375000	3.043228	1.421875	3.089337	1.328125
2.985591	1.328125	3.043228	1.406250	3.089337	1.296875
2.985591	1.312500	3.043228	1.375000	3.089337	1.281250
2.985591	1.296875	3.043228	1.359375	3.100865	1.640625
2.985591	1.281250	3.043228	1.343750	3.100865	1.625000
2.985591	1.265625	3.043228	1.328125	3.100865	1.562500
2.985591	1.234375	3.043228	1.187500	3.100865	1.546875
2.985591	1.171875	3.054755	1.640625	3.100865	1.515625
2.997118	1.453125	3.054755	1.625000	3.100865	1.500000
2.997118	1.437500	3.054755	1.593750	3.100865	1.484375
2.997118	1.421875	3.054755	1.500000	3.100865	1.468750
2.997118	1.406250	3.054755	1.484375	3.100865	1.421875
2.997118	1.390625	3.054755	1.453125	3.100865	1.406250
2.997118	1.375000	3.054755	1.437500	3.100865	1.375000

2.997118	1.250000	3.054755	1.421875	3.100865	1.359375
2.997118	1.187500	3.054755	1.406250	3.100865	1.343750
3.008646	1.593750	3.054755	1.390625	3.100865	1.328125
3.008646	1.500000	3.054755	1.328125	3.100865	1.296875
3.008646	1.484375	3.054755	1.312500	3.100865	1.281250
3.008646	1.421875	3.054755	1.296875	3.100865	1.265625
3.008646	1.406250	3.054755	1.281250	3.112392	1.531250
3.008646	1.375000	3.054755	1.265625	3.112392	1.500000
3.008646	1.328125	3.054755	1.250000	3.112392	1.484375
3.008646	1.265625	3.066282	1.562500	3.112392	1.468750
3.008646	1.218750	3.066282	1.546875	3.112392	1.453125
3.008646	1.203125	3.066282	1.500000	3.112392	1.421875
3.008646	1.156250	3.066282	1.484375	3.112392	1.406250
3.008646	1.140625	3.066282	1.453125	3.112392	1.390625
3.020173	1.593750	3.066282	1.437500	3.112392	1.375000
3.020173	1.500000	3.066282	1.421875	3.112392	1.312500
3.020173	1.484375	3.066282	1.406250	3.112392	1.296875
3.020173	1.421875	3.066282	1.390625	3.112392	1.281250
3.020173	1.406250	3.066282	1.375000	3.112392	1.265625
3.020173	1.375000	3.066282	1.328125	3.123919	1.531250
3.020173	1.328125	3.066282	1.312500	3.123919	1.500000
3.020173	1.265625	3.066282	1.234375	3.123919	1.484375
3.020173	1.218750	3.066282	1.218750	3.123919	1.468750
3.020173	1.203125	3.066282	1.203125	3.123919	1.453125
3.020173	1.156250	3.077810	1.562500	3.123919	1.421875
3.020173	1.140625	3.077810	1.546875	3.123919	1.406250
3.031700	1.531250	3.077810	1.500000	3.123919	1.390625
3.031700	1.500000	3.077810	1.484375	3.123919	1.375000
3.031700	1.484375	3.077810	1.453125	3.123919	1.312500
3.031700	1.453125	3.077810	1.437500	3.123919	1.296875
3.031700	1.421875	3.077810	1.421875	3.123919	1.281250
3.031700	1.406250	3.077810	1.406250	3.123919	1.265625
3.031700	1.375000	3.077810	1.390625	3.135447	1.640625
3.031700	1.359375	3.077810	1.375000	3.135447	1.625000
3.031700	1.343750	3.077810	1.328125	3.135447	1.515625
3.135447	1.468750	3.170029	1.390625	3.216138	1.500000
3.135447	1.453125	3.170029	1.375000	3.216138	1.484375
3.135447	1.437500	3.170029	1.265625	3.216138	1.468750
3.135447	1.421875	3.181556	1.718750	3.216138	1.453125
3.135447	1.406250	3.181556	1.703125	3.216138	1.437500
3.135447	1.390625	3.181556	1.687500	3.216138	1.375000
3.135447	1.328125	3.181556	1.640625	3.227666	1.718750
3.135447	1.296875	3.181556	1.625000	3.227666	1.703125
3.135447	1.281250	3.181556	1.609375	3.227666	1.687500
3.135447	1.265625	3.181556	1.593750	3.227666	1.671875
3.146974	1.703125	3.181556	1.515625	3.227666	1.609375
3.146974	1.687500	3.181556	1.500000	3.227666	1.593750
3.146974	1.671875	3.181556	1.484375	3.227666	1.578125
3.146974	1.640625	3.181556	1.468750	3.227666	1.531250

3.146974	1.625000	3.181556	1.453125	3.227666	1.515625
3.146974	1.593750	3.181556	1.437500	3.227666	1.500000
3.146974	1.531250	3.181556	1.390625	3.227666	1.484375
3.146974	1.515625	3.181556	1.375000	3.227666	1.468750
3.146974	1.468750	3.181556	1.265625	3.227666	1.453125
3.146974	1.453125	3.193084	1.671875	3.227666	1.437500
3.146974	1.437500	3.193084	1.656250	3.227666	1.375000
3.146974	1.421875	3.193084	1.609375	3.239193	1.781250
3.146974	1.406250	3.193084	1.578125	3.239193	1.765625
3.146974	1.390625	3.193084	1.562500	3.239193	1.734375
3.146974	1.359375	3.193084	1.546875	3.239193	1.703125
3.146974	1.343750	3.193084	1.531250	3.239193	1.687500
3.146974	1.328125	3.193084	1.500000	3.239193	1.671875
3.146974	1.296875	3.193084	1.484375	3.239193	1.640625
3.146974	1.281250	3.193084	1.468750	3.239193	1.625000
3.158501	1.703125	3.193084	1.453125	3.239193	1.609375
3.158501	1.687500	3.193084	1.437500	3.239193	1.593750
3.158501	1.656250	3.193084	1.390625	3.239193	1.578125
3.158501	1.640625	3.204611	1.703125	3.239193	1.562500
3.158501	1.625000	3.204611	1.687500	3.239193	1.546875
3.158501	1.562500	3.204611	1.656250	3.239193	1.531250
3.158501	1.546875	3.204611	1.640625	3.239193	1.515625
3.158501	1.515625	3.204611	1.625000	3.239193	1.500000
3.158501	1.468750	3.204611	1.609375	3.239193	1.484375
3.158501	1.453125	3.204611	1.593750	3.239193	1.468750
3.158501	1.437500	3.204611	1.578125	3.250720	1.781250
3.158501	1.390625	3.204611	1.562500	3.250720	1.765625
3.158501	1.375000	3.204611	1.546875	3.250720	1.671875
3.158501	1.359375	3.204611	1.531250	3.250720	1.656250
3.158501	1.343750	3.204611	1.500000	3.250720	1.640625
3.158501	1.328125	3.204611	1.484375	3.250720	1.625000
3.170029	1.718750	3.204611	1.468750	3.250720	1.609375
3.170029	1.703125	3.204611	1.453125	3.250720	1.593750
3.170029	1.687500	3.204611	1.421875	3.250720	1.578125
3.170029	1.640625	3.204611	1.406250	3.250720	1.562500
3.170029	1.625000	3.216138	1.718750	3.250720	1.546875
3.170029	1.609375	3.216138	1.703125	3.250720	1.531250
3.170029	1.593750	3.216138	1.687500	3.250720	1.515625
3.170029	1.515625	3.216138	1.671875	3.250720	1.500000
3.170029	1.500000	3.216138	1.609375	3.250720	1.484375
3.170029	1.484375	3.216138	1.593750	3.250720	1.468750
3.170029	1.468750	3.216138	1.578125	3.250720	1.453125
3.170029	1.453125	3.216138	1.531250	3.250720	1.421875
3.170029	1.437500	3.216138	1.515625	3.250720	1.406250
3.262248	1.796875	3.296830	1.546875	3.342939	1.531250
3.262248	1.750000	3.296830	1.531250	3.354467	1.750000
3.262248	1.734375	3.296830	1.515625	3.354467	1.734375
3.262248	1.718750	3.296830	1.500000	3.354467	1.703125
3.262248	1.703125	3.296830	1.484375	3.354467	1.687500

3.262248	1.687500	3.296830	1.453125	3.354467	1.671875
3.262248	1.640625	3.308357	1.734375	3.354467	1.656250
3.262248	1.625000	3.308357	1.718750	3.354467	1.640625
3.262248	1.609375	3.308357	1.703125	3.354467	1.625000
3.262248	1.578125	3.308357	1.687500	3.354467	1.609375
3.262248	1.562500	3.308357	1.671875	3.354467	1.593750
3.262248	1.546875	3.308357	1.656250	3.354467	1.578125
3.262248	1.531250	3.308357	1.640625	3.354467	1.531250
3.262248	1.515625	3.308357	1.625000	3.365994	1.750000
3.262248	1.500000	3.308357	1.609375	3.365994	1.734375
3.262248	1.484375	3.308357	1.593750	3.365994	1.718750
3.262248	1.468750	3.308357	1.578125	3.365994	1.703125
3.262248	1.437500	3.308357	1.562500	3.365994	1.687500
3.273775	1.796875	3.308357	1.546875	3.365994	1.656250
3.273775	1.750000	3.308357	1.531250	3.365994	1.640625
3.273775	1.734375	3.308357	1.468750	3.365994	1.625000
3.273775	1.718750	3.308357	1.453125	3.365994	1.593750
3.273775	1.703125	3.319885	1.796875	3.365994	1.578125
3.273775	1.687500	3.319885	1.781250	3.365994	1.562500
3.273775	1.640625	3.319885	1.765625	3.365994	1.546875
3.273775	1.625000	3.319885	1.734375	3.365994	1.531250
3.273775	1.609375	3.319885	1.718750	3.365994	1.500000
3.273775	1.578125	3.319885	1.703125	3.365994	1.484375
3.273775	1.562500	3.319885	1.687500	3.377522	1.750000
3.273775	1.546875	3.319885	1.671875	3.377522	1.734375
3.273775	1.531250	3.319885	1.656250	3.377522	1.718750
3.273775	1.515625	3.319885	1.640625	3.377522	1.703125
3.273775	1.500000	3.319885	1.625000	3.377522	1.687500
3.273775	1.484375	3.319885	1.593750	3.377522	1.656250
3.273775	1.468750	3.319885	1.531250	3.377522	1.640625
3.273775	1.437500	3.319885	1.515625	3.377522	1.625000
3.285303	1.750000	3.331412	1.796875	3.377522	1.593750
3.285303	1.734375	3.331412	1.781250	3.377522	1.578125
3.285303	1.718750	3.331412	1.765625	3.377522	1.562500
3.285303	1.703125	3.331412	1.734375	3.377522	1.546875
3.285303	1.687500	3.331412	1.718750	3.377522	1.531250
3.285303	1.671875	3.331412	1.703125	3.377522	1.500000
3.285303	1.656250	3.331412	1.687500	3.377522	1.484375
3.285303	1.609375	3.331412	1.671875	3.389049	1.859375
3.285303	1.578125	3.331412	1.656250	3.389049	1.843750
3.285303	1.562500	3.331412	1.640625	3.389049	1.828125
3.285303	1.546875	3.331412	1.625000	3.389049	1.734375
3.285303	1.531250	3.331412	1.593750	3.389049	1.718750
3.285303	1.515625	3.331412	1.531250	3.389049	1.703125
3.285303	1.468750	3.331412	1.515625	3.389049	1.687500
3.285303	1.453125	3.342939	1.796875	3.389049	1.671875
3.296830	1.671875	3.342939	1.781250	3.389049	1.656250
3.296830	1.640625	3.342939	1.765625	3.389049	1.640625
3.296830	1.625000	3.342939	1.750000	3.389049	1.625000

3.296830	1.609375	3.342939	1.671875	3.389049	1.578125
3.296830	1.593750	3.342939	1.593750	3.400576	1.937500
3.296830	1.578125	3.342939	1.562500	3.400576	1.812500
3.296830	1.562500	3.342939	1.546875	3.400576	1.781250
3.400576	1.765625	3.446686	1.734375	3.515850	1.843750
3.400576	1.734375	3.446686	1.703125	3.515850	1.828125
3.400576	1.718750	3.446686	1.687500	3.515850	1.781250
3.400576	1.703125	3.446686	1.671875	3.515850	1.765625
3.400576	1.687500	3.446686	1.656250	3.515850	1.750000
3.400576	1.671875	3.458213	1.906250	3.515850	1.734375
3.400576	1.656250	3.458213	1.890625	3.515850	1.703125
3.400576	1.640625	3.458213	1.750000	3.515850	1.687500
3.400576	1.625000	3.458213	1.734375	3.515850	1.671875
3.400576	1.609375	3.458213	1.718750	3.527378	1.984375
3.400576	1.593750	3.458213	1.703125	3.527378	1.968750
3.412104	1.843750	3.458213	1.687500	3.527378	1.937500
3.412104	1.828125	3.458213	1.671875	3.527378	1.906250
3.412104	1.812500	3.458213	1.656250	3.527378	1.890625
3.412104	1.781250	3.469741	1.937500	3.527378	1.843750
3.412104	1.765625	3.469741	1.796875	3.527378	1.828125
3.412104	1.734375	3.469741	1.781250	3.527378	1.781250
3.412104	1.718750	3.469741	1.765625	3.527378	1.765625
3.412104	1.703125	3.469741	1.750000	3.527378	1.750000
3.412104	1.687500	3.469741	1.734375	3.527378	1.734375
3.412104	1.671875	3.469741	1.718750	3.527378	1.703125
3.412104	1.656250	3.469741	1.703125	3.527378	1.687500
3.412104	1.640625	3.469741	1.687500	3.527378	1.671875
3.412104	1.625000	3.469741	1.671875	3.538905	1.984375
3.412104	1.593750	3.469741	1.562500	3.538905	1.968750
3.423631	1.843750	3.469741	1.546875	3.538905	1.937500
3.423631	1.828125	3.481268	1.937500	3.538905	1.906250
3.423631	1.812500	3.481268	1.796875	3.538905	1.890625
3.423631	1.781250	3.481268	1.781250	3.538905	1.875000
3.423631	1.765625	3.481268	1.765625	3.538905	1.781250
3.423631	1.734375	3.481268	1.750000	3.538905	1.765625
3.423631	1.718750	3.481268	1.734375	3.538905	1.750000
3.423631	1.703125	3.481268	1.718750	3.538905	1.718750
3.423631	1.687500	3.481268	1.703125	3.538905	1.703125
3.423631	1.671875	3.481268	1.687500	3.538905	1.687500
3.423631	1.656250	3.481268	1.671875	3.538905	1.656250
3.423631	1.640625	3.481268	1.562500	3.550432	1.937500
3.423631	1.625000	3.481268	1.546875	3.550432	1.843750
3.423631	1.593750	3.492795	1.843750	3.550432	1.828125
3.435159	1.921875	3.492795	1.828125	3.550432	1.812500
3.435159	1.796875	3.492795	1.812500	3.550432	1.781250
3.435159	1.781250	3.492795	1.796875	3.550432	1.765625
3.435159	1.765625	3.492795	1.750000	3.550432	1.718750
3.435159	1.734375	3.492795	1.734375	3.561960	1.937500
3.435159	1.718750	3.492795	1.718750	3.561960	1.812500

3.435159	1.703125	3.492795	1.671875	3.561960	1.781250
3.435159	1.687500	3.492795	1.609375	3.561960	1.765625
3.435159	1.671875	3.504323	1.859375	3.561960	1.734375
3.435159	1.656250	3.504323	1.812500	3.573487	1.984375
3.435159	1.640625	3.504323	1.796875	3.573487	1.968750
3.435159	1.625000	3.504323	1.750000	3.573487	1.859375
3.435159	1.578125	3.504323	1.703125	3.573487	1.812500
3.446686	1.906250	3.504323	1.687500	3.573487	1.796875
3.446686	1.890625	3.515850	1.984375	3.573487	1.750000
3.446686	1.812500	3.515850	1.968750	3.573487	1.734375
3.446686	1.781250	3.515850	1.937500	3.585014	1.984375
3.446686	1.765625	3.515850	1.906250	3.585014	1.968750
3.446686	1.750000	3.515850	1.890625	3.585014	1.859375
3.585014	1.812500	3.654179	1.765625	3.884726	2.250000
3.585014	1.796875	3.665706	1.875000	3.884726	2.234375
3.585014	1.750000	3.665706	1.843750	3.896254	2.125000
3.585014	1.734375	3.665706	1.828125	3.896254	2.109375
3.596542	1.921875	3.665706	1.812500	3.896254	2.078125
3.596542	1.906250	3.677233	1.875000	3.942363	2.328125
3.596542	1.890625	3.677233	1.843750	3.942363	2.312500
3.596542	1.812500	3.677233	1.828125	3.942363	2.125000
3.596542	1.796875	3.677233	1.812500	3.942363	2.109375
3.596542	1.781250	3.688761	2.062500	3.942363	2.093750
3.596542	1.765625	3.688761	2.015625	3.965418	2.250000
3.608069	2.046875	3.688761	2.000000	3.965418	2.234375
3.608069	2.031250	3.688761	1.984375	3.965418	2.203125
3.608069	2.000000	3.688761	1.968750	3.965418	2.125000
3.608069	1.953125	3.688761	1.921875	3.965418	2.109375
3.608069	1.921875	3.688761	1.875000	3.976945	2.250000
3.608069	1.906250	3.688761	1.859375	3.976945	2.234375
3.608069	1.890625	3.688761	1.843750	3.976945	2.093750
3.608069	1.812500	3.688761	1.828125	3.988473	2.250000
3.608069	1.796875	3.700288	2.015625	3.988473	2.234375
3.608069	1.734375	3.700288	2.000000	3.988473	2.093750
3.608069	1.718750	3.700288	1.953125	4.000000	2.562500
3.619597	2.015625	3.700288	1.906250	4.000000	2.484375
3.619597	2.000000	3.700288	1.890625	4.000000	2.468750
3.619597	1.953125	3.700288	1.859375	4.000000	2.453125
3.619597	1.906250	3.700288	1.796875	4.000000	2.406250
3.619597	1.890625	3.711816	1.921875	4.000000	2.390625
3.619597	1.796875	3.711816	1.906250	4.000000	2.375000
3.619597	1.781250	3.711816	1.890625	4.000000	2.218750
3.619597	1.765625	3.711816	1.859375		
3.619597	1.750000	3.723343	1.937500		
3.631124	2.015625	3.723343	1.859375		
3.631124	2.000000	3.723343	1.796875		
3.631124	1.953125	3.734870	1.937500		
3.631124	1.906250	3.734870	1.859375		
3.631124	1.890625	3.734870	1.796875		

3.631124	1.796875	3.746398	2.062500
3.631124	1.781250	3.746398	2.046875
3.631124	1.765625	3.746398	2.031250
3.631124	1.750000	3.746398	1.921875
3.642651	2.000000	3.746398	1.812500
3.642651	1.921875	3.757925	2.046875
3.642651	1.875000	3.757925	2.031250
3.642651	1.843750	3.769452	1.859375
3.642651	1.828125	3.780980	1.859375
3.642651	1.812500	3.792507	2.125000
3.654179	2.140625	3.792507	2.109375
3.654179	2.078125	3.792507	1.984375
3.654179	2.015625	3.792507	1.968750
3.654179	2.000000	3.792507	1.953125
3.654179	1.984375	3.804035	2.046875
3.654179	1.968750	3.804035	2.031250
3.654179	1.921875	3.815562	2.046875
3.654179	1.875000	3.815562	2.031250
3.654179	1.859375	3.850144	2.078125
3.654179	1.843750	3.861671	2.140625
3.654179	1.828125	3.873199	2.250000
3.654179	1.781250	3.873199	2.234375

REFERENCES

1. D. Luo. The road from biology to materials. Materials Today, 6, 38-43 (2003)
2. Y. Li, Y.D. Tseng, S.Y. Kown, L. d'Espaux, J.S. Bunch, P.L McEuen and D. Luo. Controlled assembly of dendrimer-like DNA. Nature Materials, 3, 38-42 (2004)
3. Y. Li, Y. Cu and D. Luo, DNA fluorescence nanobarcodes for multiplexed pathogen detection, Accepted; Nature Biotechnology (2005)
4. http://www.normankoren.com/light_color.html
5. http://www.geologie.ens.fr/DOC/MATLAB6
6. http://csep1.phy.ornl.gov/xv/xv.13.html
7. K. Jain. Nanodiagnostics: application of nanotechnology in molecular diagnostics. Expert Review of Molecular Diagnostics, 3 (2), 153-161 (2003)
8. R. Freeman, P. Raju, S. Norton, I. Walton, P. Smith, L. He, M. Nathan, M. Sha, and S. Penn. Use of nanobarcodes particles in bioassays. Methods in Molecular Biology, 303, 73-81 (2005)