

Area and Eccentricity of Fibroblasts During Trypsinization

Kyle Thomas
(Dated: May 11, 2015)

Area and eccentricity of mouse-embryo fibroblast cells are both measured as the cells release the top surface of collagen gel. Image sets of the cells are collected at regular time-intervals during and after detachment. Software was written to segment selected cells and measure both the area and eccentricity of each segmented cell. Some code is included in appendices as well as a link to all of the code.

EXPERIMENT METHODS

Mouse embryo fibroblasts exhibiting green fluorescent protein (cell line NIH3T3/GFP) are placed atop a collagen-gel and suspended in growth medium(see the diagram in Figure 1). The fibroblasts attach to the surface of the collagen at 37°C for 15 to 30 hours in a CO₂-regulated incubator. The cells were not seen to undergo mitosis.

Cells are imaged every regularly beginning 40 minutes before trypsinization via TrypLE Select. The images are collected with a confocal microscope using a 70/30 transmission/reflection beam-splitter. The images are collected in image sets called z-stacks.

A z-stack is a sequence of images taken in quick succession. After the first, lowest image, each image is focused a few microns higher than the preceding image. Imaged in this way, z-stacks essentially provide a "snapshot" of the state of specific cells within the sample.

The stage is adjusted before imaging a sample so that the z-stacks image from ten microns below the surface of the collagen to ten microns above the surface of the collagen. Ten microns is approximately the radius of these cells when not attached to the collagen. Z-stacks are captured every five to fifteen minutes.

Imaging the cells occurs by irradiating the sample with a 488nm laser. The green fluorescent proteins fluoresce at a different wavelength than the 488nm excitation light,

thereby distinguishing fluorescence from reflection. Light from 500nm to 550nm is observed for emissions of the GFP.

The fibers were also observed for alignment phenomena which were not seen. To image the fibers, the sample is excited by a 532nm laser and wavelengths in the spectrum 531nm to 536nm are observed for reflection and transmission of the incident light off of the fibers. The observation spectrum imaged for the fibers was found by visually determining the spectrum in which the fibers appear most distinct.

Light detected from the fibers is maximized by observing a spectrum symmetric about the wavelength of the incident light, say 526nm to 536nm. However, such a spectrum has a higher signal to noise ratio because more of the light emitted by the GFP is detected below 531nm.

Both sets of images are acquired with a 30% reflection, 70% transmission beamsplitter. Using the same beamsplitter for both images means time is not spent changing the beamsplitter between images. This time savings between frames allows a z-stack to be captured closer to instantaneously. An instantaneous "snapshot" of the z-stack is ideal.

During imaging, the growth-medium surrounding the cells is rinsed with 1XPBS twice. The PBS is next replaced with TrypLE Select. The TrypLE Select breaks the proteins binding the cells to the collagen-matrix.

The proteins that bind the cells to the collagen are on bulges in the cell-membrane. These protrusions are called filopodia. After the bonds break, these filopodia flatten back into the cells.

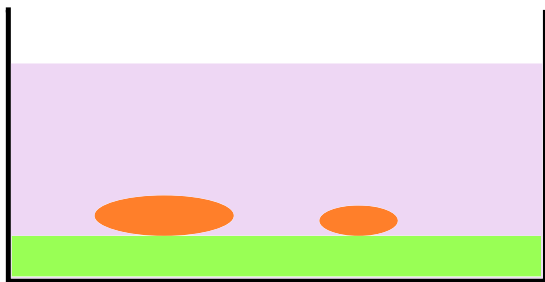
This flattening of the filopodia leaves the cells somewhat spherical. The spherical fibroblasts appear as circles when seen through the microscope. While the cells return to a spherical shape, their lack of sphericity can be measured as eccentricity, as in the eccentricity of an ellipse.

IMAGE PROCESSING METHODS

A single experiment produces a folder containing all the z-stacks acquired during the experiment. Each z-stack is comprised of several images, as described before. The image files for a z-stack are combined into a single TIFF using Matlab. A maximal intensity projection is used to transform all images comprising a z-stack into a single image.

The Matlab function mip creates a maximal intensity projection for each z-stack. This maximal intensity projection is a single image. The maximal intensity projection is made by using the value of the brightest pixel at

FIG. 1. Orange cells submerged in violet growth medium atop green collagen are diagrammed.



position (x_1, y_1) of every image in the image's z-stack. An example MIP-creation process is shown in Figure 2.



FIG. 2. The mip function is illustrated projecting simple example images.

The function maskmaker is used to draw a region loosely enclosing a single cell at its most stretched out. Maskmaker sets pixels outside the region to black. This creates a mask loosely surrounding the cell of interest. Applying this mask leaves the user-selected cell as the largest object in the image so that the segmentation of the selected cell can proceed.

The function stackmasker applies this mask to all images in the stack. All masked images contain the entire cell of interest. The mask is drawn around the cell when the filopodia are stretched out to ensure the whole cell is visible in later images as the cell's filopodia recede.

Masking works by ensuring that the cell of interest is the largest object shown. A mask is made loosely enclosing a single cell so that the user can specify a certain cell for software to outline precisely. Cells are precisely outlined by software to save time and to ensure uniformity in drawing outlines.

Filling the objects in the image proceeds as follows. The brightness of each masked image is multiplied by a constant factor. This factor is 15 by default for the trypsinization images. Some z-stacks were brightened by a factor of more than 15 to facilitate masking entire cells.

The Matlab Image Processing Toolbox function `im2bw` uses a brightness threshold to create a black and white image from a grayscale image. Pixels brighter than the threshold value are converted to white while other pixels are turned to black. The threshold used to create binary (black and white) images is 0.15.

Several experiments show multiple cells in each image. For example, consider an image that contains cells A and B. Each cell's shape and eccentricity can be measured separately.

A cell is measured by creating a separate binary set of images for that cell. The cell is segmented by ensuring it is the only object visible in its image-set.

Cell A is loosely masked by the user so that it's the largest object visible. Cell A is segmented in the images. These images showing only cell A are saved. The area and eccentricity of cell A is measured from these images. This process is repeated to measure cell B.

Image segmentation yields multipage-tiff-files, where each page is a 1024 pixel by 1024 pixel image. All images in a file show only the same particular cell. The images are taken 20 minutes apart so that each file watches a different cell as it releases the collagen substrate.

APPENDICES

Appendix A: Experiment Protocol

The protocol for three collagen-samples of concentration 1.5 and gelled at 21°C is as follows:

1. Create the collagen-solution.
 - (a) Total volume to be made is $V = (90\mu\text{L per sample})(3 \text{ samples})(150\%) = 405\mu\text{L}$.
 - (b) Combine $40.5\mu\text{L}$ 10X PBS, $14.2\mu\text{L}$ NaOH, and $276.3\mu\text{L}$ growth-medium.
 - (c) Shake this solution at 400rpm for 20 seconds.
 - (d) Store at 6°C for 10 minutes.
 - (e) Add $71\mu\text{L}$ collagen homogeneously.
 - (f) Stir collagen into solution with a $100\mu\text{L}$ pipette tip.
 - (g) Pipette $90\mu\text{L}$ collagen-solution into each of the 3 samples a, b, and c.
2. Wrap each sample in film and let gelate at 21°C for two hours.
3. Place growth-medium on all samples and place cells on sample a.
4. Place cells on sample b four hours later.
5. Place cells on sample c four hours later.
6. Using confocal fluorescence microscopy, capture a $10\mu\text{m}$ tall z-stack of images of fibers and cells, separately, every 20 minutes for four hours.
 - (a) After the first z-stack, replace the growth-medium on the sample with 1X PBS.
 - (b) After the second z-stack, replace the PBS with fresh PBS.
 - (c) After the third z-stack, replace the PBS with 1X TrypLe™ Select.
7. When sample a is finished imaging, image sample b in the same way as sample a was imaged.
8. When sample b is finished imaging, image sample c in the same way as sample a was imaged.

Appendix B: Code

```

1 % this script segments each object in a tiff-stack in its own stack
2 % by asking the user to mask a region constaining only one object.
3 % user decides how much to brighten and threshold the image for
4 % segmentation
5 %directory = ['/windows/Users/thomasky/Research/2D-fibertraction/'...
6 % '_prep-to-mask-and-binarize/good-images-no-beads'];
7 directory = ['C:\Users\thomasky\Research\2D-fibertraction\'...
8 '_prep-to-mask-and-binarize\good-images-no-beads'];
9 outputPath = fullfile(directory, '..', '..', 'binary');
10 threshold = 0.15;
11 defaultBrighteningFactor = 15;
12 folders = ls(directory);
13 %% attempt to resume from the folder that the program left off at
14 %~~~~~commented for scripting~~~~~
15 % if exist('k','var')
16 %     resume = input(sprintf('Enter the folder number to start'...
17 %         ' with or press enter to attempt\nresuming where the'...
18 %         ' program left off.\n'));
19 %     if isempty(resume)
20 %         startingfolder = k;
21 %     else
22 %         startingfolder = resume;
23 %     end
24 % else
25 %     startingfolder = 1;
26 % end
27 for k=startingfolder:length(folders(:,1))
28
29
30
31 %% read all folders in directory and skip folders with no tiffs
32 stackDirectory = fullfile(directory, strtrim(folders(k,:)));
33 if size(dir(fullfile(stackDirectory, '*.tif')),1)==0 &&...
34     isdir(stackDirectory)
35     continue
36 end
37
38 %% read stack in current folder
39 disp(['FOLDER #' num2str(k) ': ' folders(k,:)]);
40 stack = stackreader(stackDirectory, '*.tif');
41
42 %% Loop through binarizing stack until ok
43 %~~~~~commented for scripting~~~~~
44 % if exist('brightenedStack','var')
45 %     clear('brightenedStack');
46 % end
47 % ok = 0;
48 % while ~isempty(ok)
49 %
50 %     % get brightening factor from user or use default
51 %     if ok ~= 0
52 %         brighteningFactor = ok;
53 %     end
54 %     if ~exist('brighteningFactor','var')
55 %         brighteningFactor = defaultBrighteningFactor;
56 %     end
57 %~~~~~commented for scripting~~~~~
58 % brighten, binarize, and fill the stack
59 bwStack = zeros(size(stack));
60 for page=1:size(stack,3)
61     bwStack(:, :, page) = ...
62         imfill(im2bw(brighteningFactor*stack(:, :, page)), ...
63             'holes');
64 end
65
66 % show stack and ask if ok, not ok => redo w/ new parameters

```

```

69 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%commented for scripting%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       implay(bwStack);
71 %       ok = input(['PRESS ENTER IF B&W STACK IS OK OR ENTER NEW '...
%                   'BRIGHTENING FACTOR: ']);
73
end
75 %% let user mask region containing mth object amd decide whether to
% save the selected object's stack
77
% ask how many objects in the current stack will be masked
79 numobjects = input(sprintf(['How many objects will be'...
% segmented in this stack?\n']));
81
% for each object, let user create mask
83 for m = 1:numobjects
    disp(['MASKING OBJECT #' num2str(m) ...
85         ' IN STACK ' folders(k,:) ]);
    mask = maskmaker(bwStack);
87    while 1==1
        appendObjectToFilename = ['object-' num2str(m)];
89        maskedstack = stackmasker(bwStack,mask,threshold);
        for page=1:size(maskedstack,3)
91            maskedstack(:,:,page) =...
                ExtractNLargestBlobs(maskedstack(:,:,page),1);
93        end

95        % show masked stack. ask if ok. ok => save 1-object-stack
        implay(maskedstack);
97        ok = input(sprintf(['Either enter 1 to save'...
% the binary stack, press enter to skip,\nor enter'...
% anything else to tinker.\n']));
99        if ok == 1
            disp(['WRITING BINARY STACK:' folders(k,:) '.tif']);
            stackwriter(maskedstack,outputPath,...
103                ['binary-' folders(k,:) ],...
                '',appendObjectToFilename);
            disp(['SUCCESSFULLY WROTE THE STACK NAMED binary-',...
105                folders(k,:) ,appendObjectToFilename, '.tif' ])
            break
107        elseif isempty(ok)
            break
109        else
            tinker
111        end
113    end
    continue
115 end
end

```

Listing 1. segmenter.m

```

folder = 'C:\Users\thomasky\Research\2D-fibertraction\binary';
2 files = ls(fullfile(folder, '*.tif'));
for file = 1:size(files,1)
4
    % ask for time interval between z-stacks
6    stack = stackreader(folder, files(file, :));

8    % get time, area, and eccetricity values of object for each page
    disp('#####');
10    disp(['PROCESSING FOLDER #' num2str(file) files(file, :)]);
    disp('#####');
12    timeInterval = input('\nTIME BETWEEN Z-STACKS: ');
    timeOffset = input(['NUMBER OF Z-STACKS BEFORE '...
14                        'TRYPLE SELECT WAS ADDED: ']);

    time = zeros(size(stack,3),1);
16    area = zeros(size(stack,3),1);
    eccentricity = zeros(size(stack,3),1);
18    for page = 1:size(stack,3)
        time(page) = (page - timeOffset - 1)*timeInterval
20        info = regionprops(stack(:,:,page), 'Area', 'Eccentricity');
        for region = 1:size(info,1)
22            if info(region).Area ~= 0
                area(page) = info(region).Area
24            end
            if info(region).Eccentricity ~= 0
26                eccentricity(page) = info(region).Eccentricity
            end
28        end
    end
29    % store time, area, and eccetricity values of object for each page
    data = cat(2,time,cat(2,area,eccentricity));
32
    % write time, area, and eccentricity to csv
34    csvFilename = fullfile(folder,[ files(file, :) '.csv' ]);
    dlmwrite(csvFilename,data,'newline','pc')
36    disp(sprintf(['WROTE DATA TO FILE ' ]));
38 end

```

Listing 2. regionPropsScript.m

```

% creates separate scatter plots of data from csv
2
% initialize variables
4 csvDirectory = 'C:\Users\thomasky\Research\2D-fibertraction\binary';
  csvFiles = ls(fullfile(csvDirectory, '*.csv'));
6
% initialize figures and set each to hold
8 eccentricityDotPlot = figure('Name', ['Relative Change in' ...
                                     'Eccentricity vs. Time'], ...
10                                     'NumberTitle', 'off');
  hold all;
12 eccentricityPlot = figure('Name', 'Eccentricity vs. Time', ...
                            'NumberTitle', 'off');
14 hold all;
  areaDotPlot = figure('Name', 'Relative Change in Area vs. Time', ...
16                      'NumberTitle', 'off');
  hold all;
18 areaPlot = figure('Name', 'Area vs. Time', 'NumberTitle', 'off');
  hold all;
20
for csvFileIndex = 1:size(csvFiles,1)
22     csvFile      = csvFiles(csvFileIndex,:);
     csvData      = csvread(fullfile(csvDirectory, csvFile));
24
     time         = csvData(:,1);
26     area         = csvData(:,2);
     eccentricity  = csvData(:,3);
28
     % calculate the time derivative of relative area
30     areaDot = zeros(length(area)-1,1);
     for areaDotIndex = 1:length(areaDot)
32         areaDot(areaDotIndex) = ...
            (area(areaDotIndex+1)-area(areaDotIndex));%/ ...
34 %     area(1);
     end
36
     % calculate the time derivative of relative eccentricity
38     eccentricityDot = zeros(length(eccentricity)-1,1);
     for eccentricityDotIndex = 1:length(eccentricityDot)
40         eccentricityDot(eccentricityDotIndex) = ...
            (eccentricity(eccentricityDotIndex+1) ...
42             -eccentricity(eccentricityDotIndex));%/ ...
     %     /eccentricity(1);
44     end

46     figure(areaPlot);
48     scatter(time, area, '+');
     figure(eccentricityPlot);
50     scatter(time, eccentricity, '*');
     figure(areaDotPlot);
52     scatter(time(1:end-1), areaDot, '.');
     figure(eccentricityDotPlot);
54     scatter(time(1:end-1), eccentricityDot, 'x');
end

```

Listing 3. plotterScript.m