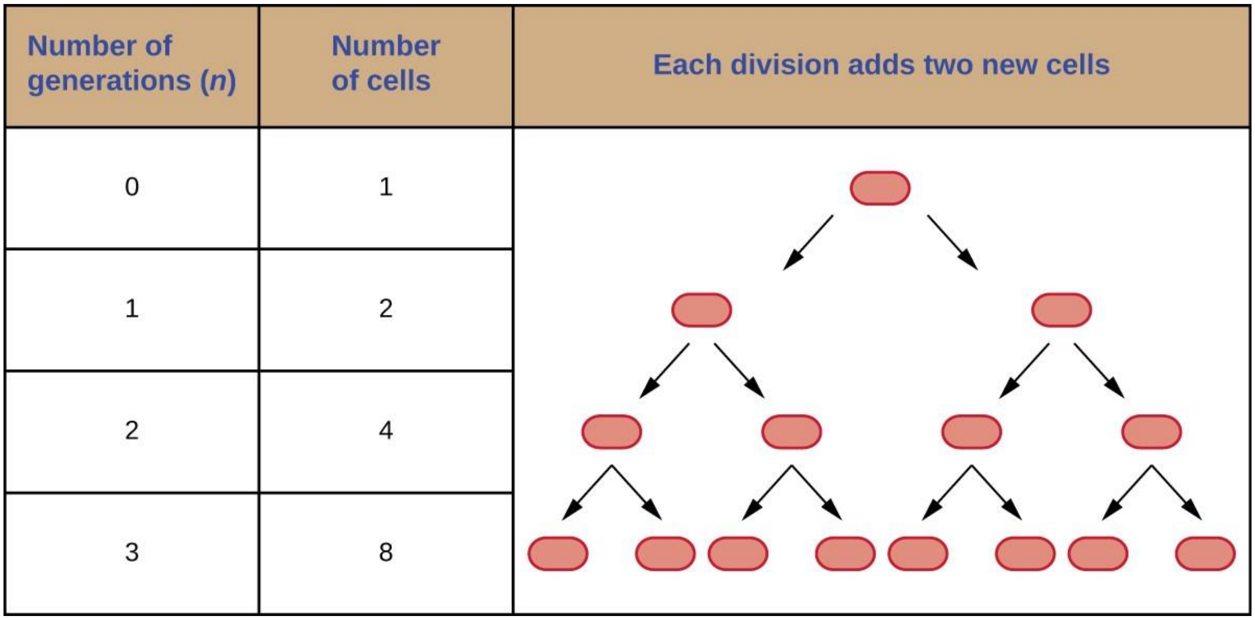
Matlab Tutorial: Measuring cell growth rate

James Boedicker, Hernan Garcia, Manuel Razo Mejia, Griffin Chure, Elizabeth Eck, Meghan Turner

Introduction

Reproduction is an essential process of life. In many single-celled organisms such as bacteria, reproduction occurs via binary fission, in which a mother cells splits in two creating daughter cells. This process of cell division occurs at a characteristic time scale, in the range of 10 minutes for the fastest growing bacteria such as *Pseudomonas natriegens* to a day, or even longer, for slow growing cells such as *Mycobcaterium tuberculosis*. The growth rate is influenced by many factors, including temperature, nutrient availability, and interactions with other cells. Drugs such as antibiotic also influence growth rates.



Number of generations (*n*)	Number of cells	Each division adds two new cells
0	1	
1	2	
2	4	
3	8	

Figure 1: Bacteria reproduce through cell division, doubling the number of cells in each new generation. Picture from https://courses.lumenlearning.com/microbiology/chapter/how-microbes-grow/.

There are many real-life situations in which accurate information about the growth rate of a cell is desirable, ranging from bacterial infections to food spoilage. In this tutorial we will measure cell growth rate using the microscope. As in Figure 2, we will take a movie of a single cell dividing multiple time to form a small colony, and in the analysis of this movie calculate the average doubling time of the cell. The following sections discuss the experimental procedures for setting up an overnight growth movie and image analysis techniques used to track the number of cells over time. The last section explores how cell growth can be simulated using the simple yet powerful numerical tool known as Euler's method.

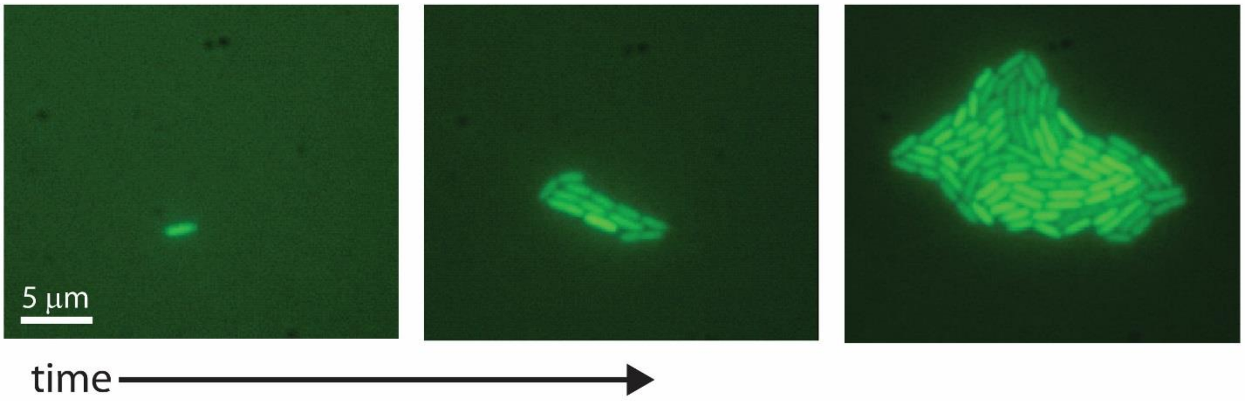


Figure 2: Optical microscopy measures the growth of *Escherichia coli* expressing GFP from a single cell to a small colony with nearly 100 cells.

Experimental protocol

1. Grow a culture of *E. coli* overnight in LB media at 37^{o}C and 200 rpm.
2. The next day, dilute the cells 1000 fold into M9 minimal media and regrow the culture to early exponential phase, meaning optical density at 600 nm and 1 cm pathlength of 0.1-0.5.
3. Make an agarose gel pad. Mix 0.1 g agarose with 10 mL of media. Microwave to melt dissolve the agarose for about 30 seconds.
4. Pipette 0.5 mL of agarose solution, while still warm, onto a 22 x 22 mm^2 glass coverslip. Quickly place a top coverslip of the same size on top of the droplet of agarose solution. The agarose solution should remain in between the two coverslips.
5. Wait 10 minutes for the gel to set. Slide off the top coverslip and using a razor blade cut the gel into small squares (approximately 5 mm x 5 mm).
6. Dilute the cell culture at 0.1-0.5 OD_{600} approximately 10 fold in fresh media. The density should be adjusted such that there is approximately 1 cells per microscope field of view at 100X magnification.
7. Pipette 1 μL of cell culture onto the top of the gel pad. Insert the gel with the droplet of culture onto a coverslip bottom glass dish.
8. Bring the dish with lid to the microscope. Place the dish in the microscope incubator set to 37oC. (Note the microscope incubator should be brought to 37oC at least 3 hours prior to the measurement.) Uncover the dish and wait 5 minutes while the temperature of the gel to equilibrate with the incubator chamber. Place the lid on the dish and wrap parafilm around the sides to prevent moisture loss during the overnight measurement.
9. Set up the dish for overnight imaging at 100X magnification. First find the focal plane of the cells in phase microscopy. This can be a difficult step. The condenser height must be correct to see the cells easily in phase microscopy. Also ensure that the phase ring matches the phase ring needed for your objective. One trick to finding the focal plan containing the cells is to go to the center of the gel pad (center in X, and Y), put the microscope objective all the way at the bottom, and slowly bring the objective up while looking at the sample through eye piece. The cells should come into focus.

If no cells are seen, change the position of the stage (X and Y position) and try again. If cells are still not visible several steps may have gone wrong. Are the cells on the wrong side of the gel pad? Only cells squished up against the coverslip can be observed at 100X microscopy. Is your cell solution too dilute? If there are only a few cells on the pad, it will be difficult to find them. Try making a new pad with a less diluted culture of cells.

10. Once cells are located, scan the plate by moving the stage in the X and Y directions. Find a field of view that contains only a single cell. Center that cell in the field of view. Set up time lapse microscopy, taking images every 15 minutes for 10 hours. Data can be taken in either phase contract microscopy or fluorescent microscopy. If using fluorescent microscopy, it is a good idea to take a test image first to ensure acquisition times are sufficient to separate cells from the background auto fluorescence. It is a good idea to take data from 2-3 positions on the plate containing a single cell.

    For overnight microscopy drift in the sample can be a challenge. To prevent drift in X and Y directions securely attach your plate to the stage (for example by using stage clips or a magnetic bar.) To prevent drift in the Z direction, an autofocusing system is very beneficial. Two examples of autofocusing systems are Nikon's Perfect Focus and Zeiss' Definite Focus.

Image analysis

The strategy for image analysis is to flip through each frame of the growth movie and identify the pixels in each image belonging to bacterial cells. Pixels belonging to cells can be identified in either phase contract images (dark pixels belong to cells, light pixels are background) or fluorescence images (light pixels belong to cells and dark pixels are background). Identifying cells in images is much easier in cells expressing high levels of fluorescent proteins. The code below is written for counting pixels belonging to fluorescent cells, but it can be adapted for phase contrast growth movies.

The strategy for identifying pixels belonging to cells will be to apply a simple threshold to each image. We load the images one at a time, ask if each pixel in the image is great than or less than the threshold value, and count the total number of pixels greater the threshold (i.e. pixels belonging to cells).

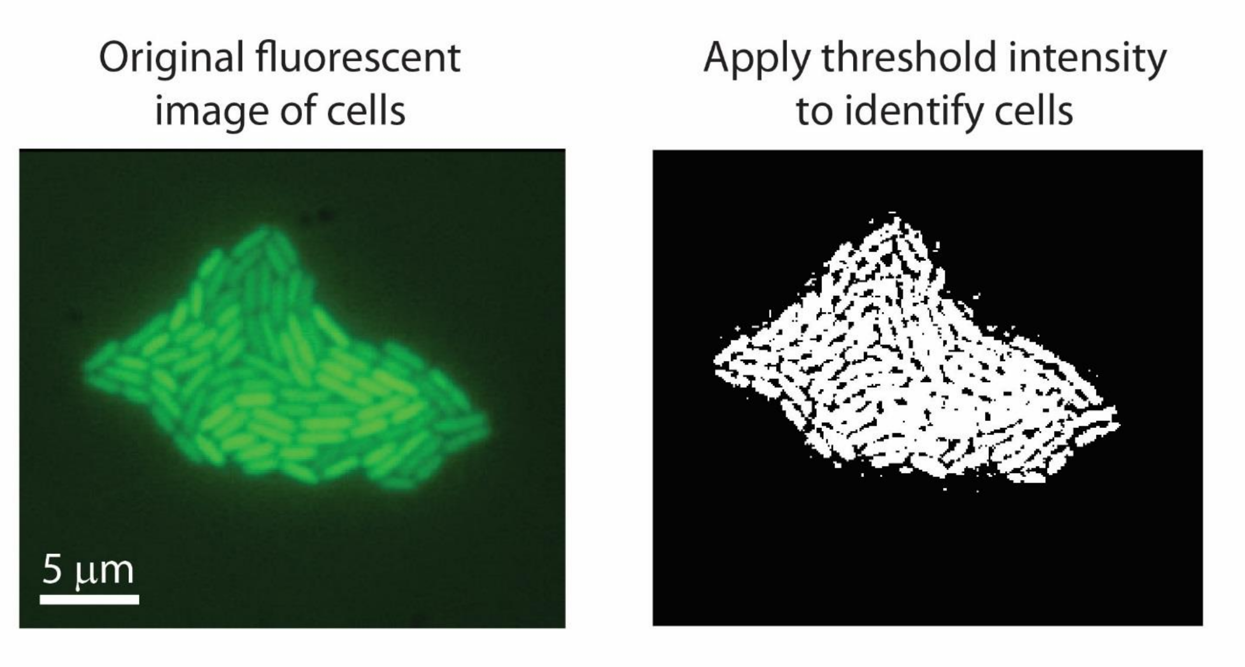

Original fluorescent image of cells
5 µm
Apply threshold intensity to identify cells

Figure 3: Fluorescent images of cells growth over time can be analyzed to identify the size of the colony at each time point. Since pixels containing cells have a higher intensity than the dark background, a simple threshold intensity can be used to identify the cells in the image.

Here is the code:

```
% look at a picture of a cell
colony = imread('colony_growth_01.tif');
imshow(colony)

% scale the image to the min and max of the pixel values
imshow(colony,[])

% automatically find the names of all images in a folder
images = dir('*.tif');

images

images(1).name

% to find a threshold value to identify the cells
colony = imread(images(1).name);
```

```matlab
imshow(colony,[])

% look at the pixel values, use pixel region
imtool(colony,[])

% a good threshold value is 150
% see if the threshold
threshold = 140;
colony_thresh = colony > threshold;

imshow(colony_thresh)

% estimate the area of the colony
% by adding up the pixels
sum(sum(colony_thresh))

%%%% Let the students try by themselves
% write a loop to analyze all 61 images
% 1. read in the image
% 2. apply the threshold
% 3. calcuate the area
% 4. record the area in a vector called 'area'


% first read in the images by number
i = 1;

colony = imread(images(i).name);

% save the data in a vector called area
area = [];

for i = 1:61
colony = imread(images(i).name);
colony_thresh = colony>threshold;
area(i) = sum(sum(colony_thresh));
end

% plot area vs. time
% create a vector of time
% pictures are 5 minutes apart
time = 0:5:300; % time in minutes

plot(time,area)

% convert area to number of cells
```

```
num_cells = area / area(5);

% the data can be fit for the rate constant r (1/min) and the
% related quantity of the doubling time
fitk = polyfit(time, log(num_cells),1);

fitk

% calculate the doubling time
log(2) / fitk(1)
```

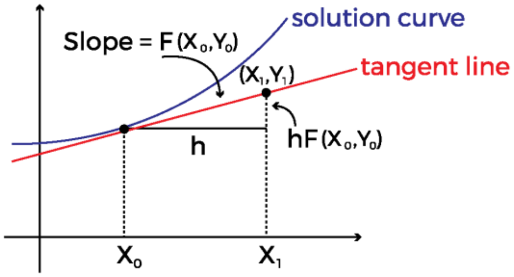
Modeling cell growth using the Euler's method

In addition to analyzing the growth rate of real cells in experiments, it is useful to simulate the growth of a colony from a single cell. For the case of a cell doubling at a regular interval (splitting in two every 30 minutes for example), growth follows the equation,

$$\frac{dN}{dt} = r \cdot N(t), \qquad \text{(Equation 1)}$$

where r is the rate constant in 1/min and N(t) is the number of cells at time t. The rate constant r, is related to how often a cell splits in two.

Equation 1 is a differential equation describing the rate of cell division. To translate this rate to the number of cells in the colony at any point in a time we will use a simple yet powerful method to simulate the change in the number of cells over time. This method is called Euler's Method. Euler's method is an algorithm for integrating the dynamics of a system given one or more differential equations describing the rate of change of system components (in our case cells), and an initial condition (how many cells we start with).



Calcworkshop.com

Figure 4: Euler's method uses the slope to estimate how the value of Y changes with X. At a starting location, (X_0,Y_0), the differential equation for dY/dX is used to predict the new Y value, Y_1, at the new X value, X_1. Euler's method follows the equation, $Y_1 = Y_0 + dY/dX * h$, where h is the change in X, X_1-X_0. This procedure is iterated to estimate Y=f(X). As shown, the estimate using Euler's method is not equivalent to the exact solution of the differential equation dY/dX. Reducing the step size, h in the image, reduces this error.

Below is a Matlab code that implements Euler's method to simulate the growth of a bacterial colony from a single cell.

```
% use the Euler method to predict the number of cells
% over time

% define some constants and initial values
N1 = 1; % start with 1 cells at t0
r = 1/30; % rate constant of cell growth in 1/min

dN_dt = N1 * r;
dt = 30; %time in min

% calculate the number of cells after 30 minutes
N2 = N1 + dN_dt * dt

% is there a faster way to do the same calculation over and
over? Write a loop in the code that does it automatically!

% first define how much time passes in between each cycle or
iteration of the loop
dt = 1; % 1 min

% define a vector N to keep track of how many cells
N = [];
% How many cells do we start with at the first time point
N(1) = 1;

% The loop will go between timepoint 2 and timepoint 500.
for t = 2:500
% Use the same step as above to calculate how many cells there
will be after dt minutes based on how many cells there were at
the previous timepoint
    N(t) = N(t-1) + N(t-1)*r * dt;
end

% To plot out results, define a vector of time
time = [1:1:500]; % in minutes

% make a plot
plot(time,N)
xlabel('time (min)')
ylabel('number of cells')
```