```
%All the tif files should be in the same directory "Particles" separated
%into folders 0h, 12h, 24h, and so on
clear;
clc;

% It is currently configured for Chlorella sp. algae
% If you would like to count a different type of cell, please change the
% radius to the following for these cell types:

%Chlorella sp. - 4
%Saccharomyces cerevisiae - 5

%If a user wishes to configure the code for a cell type not listed, measure
%the radius of at least 25 of the desired cell in pixels at the desired
%magnification and use the mean for cell_r below

%cell radius [pixels]:
cell_r = 4;
%outer radius of particle [pixels]:
particle_actual_r = 70;

%2-D Matrix containing hour * number of cells for each particle
hour_particle_cells = zeros(9,10000);
%Have vector for the number of particles in each hour
particles_for_hour = zeros(9,1);

%% Loop from initial to 96 hours:

%Load all the directories:
hour = 0;
while hour < 97

    current_directory = strcat('Particles/', num2str(hour), 'h/');

    %place all images for current time in images, will be alternating
    %bright field and Cy5

    images = dir([current_directory '*.tif']);

    %get sequence number of first image
    first_image = images(1).name;
    first_image = erase(first_image, "seq");
    first_image = erase(first_image, ".tif");
    first_image = str2num(first_image);

    %get number of last image
    [folder_size,~] = size(images);
    last_image = folder_size + first_image - 1;

    %have indices for hour and particle
    hour_index = (hour / 12) + 1;
    particle_index = 1;

    %cycle through all bright field - Cy5 pairs:
    %go through number of images / 2 times
    for seq = first_image:2:last_image
        %% Obtain particles from BF image

        %get bright field sequence # and Cy5 sequence number
        BF_number = seq;
        Cy5_number = seq + 1;

        %Load BF Image and Cy5 image
        BF_path = strcat(current_directory, images(BF_number - first_image +
1).name);
        BF_image = imread(BF_path);

        Cy5_path = strcat(current_directory, images(Cy5_number - first_image
+ 1).name);
        Cy5_image = imread(Cy5_path);

        %obtain array of particle centers and radii from BF image:
        [particle_c,particle_r] = imfindcircles(BF_image, [particle_actual_r
particle_actual_r + 25], 'ObjectPolarity', 'dark', ...
        'Sensitivity', 0.982);


        %% Now find all cells clusters using Cy5 image

        %convert Cy5 image to binary for blob analysis (need to tweak num)
        binaryCy5 = Cy5_image > 15000;

        %have a secondary Cy5 binary image to show even brighter (dense
        %regions) which will count as double (like a volume) set slightly
        %lower than maximum intensity for image:
        max_intensity = prctile(Cy5_image(:), 99.5) * 0.95;


        brighterCy5 = Cy5_image > max_intensity;

        min_blob_area = floor(pi*2^2);
        max_blob_area = floor(pi*300^2);

        cell_blob = vision.BlobAnalysis( ...
            'OutputDataType', 'single', ...
            'MinimumBlobArea', min_blob_area, ...
            'MaximumBlobArea', max_blob_area, ...
            'BoundingBoxOutputPort', true, ...
            'LabelMatrixOutputPort', true, ...
            'MaximumCount', 1500);

        %obtain areas, centroids, and bounding box of all cell blobs
        [cell_areas, cell_centers, cell_bboxes, cell_matrix] =
step(cell_blob,binaryCy5);

        %assign each blob to the number of the particle it is located within
        for i = 1:size(cell_areas)
            for j = 1:size(particle_c)
                %check if blob center is within particle in both x and y
                %direction
                if cell_centers(i,1) < particle_c(j, 1) + particle_r(j, 1) &&
cell_centers(i,1) > particle_c(j, 1) - particle_r(j, 1) ...
                        && cell_centers(i,2) < particle_c(j, 2) +
particle_r(j, 1) && cell_centers(i,2) > particle_c(j, 2) - particle_r(j, 1)

                    %place blob in particle, add number of cells in blob to
                    %the hour_particle_cells matrix:

                    %number of cells in blob obtained by area:
                    cell_number = ceil(cell_areas(i,1) / (pi * cell_r^2));

                    %increment number of cells at particle by cell number
                    hour_particle_cells(hour_index,particle_index +j-1) =
cell_number + hour_particle_cells(hour_index,particle_index +j-1);
                    break;
                end
            end
        end


        %now for the bright, dense cell clusters from the later hours
        %assign each blob to the number of the particle it is located within
        bright_matrix = zeros(2048,2048);
        if hour > 60
            [bright_areas, bright_centers, bright_bboxes, bright_matrix] =
step(cell_blob,brighterCy5);

            for i = 1:size(bright_areas)
                for j = 1:size(particle_c)
                    %check if blob center is within particle in both x and y
                    %direction
                    if bright_centers(i,1) < particle_c(j, 1) + particle_r(j,
1) && bright_centers(i,1) > particle_c(j, 1) - particle_r(j, 1) ...
                            && bright_centers(i,2) < particle_c(j, 2) +
particle_r(j, 1) && bright_centers(i,2) > particle_c(j, 2) - particle_r(j, 1)

                        %place blob in particle, add number of cells in blob
to
                        %the hour_particle_cells matrix:

                        %number of cells in blob obtained by area:
                        cell_number = 2 * ceil(bright_areas(i,1) / (pi *
cell_r^2));

                        %increment number of cells at particle by cell number
                        hour_particle_cells(hour_index,particle_index +j-1) =
cell_number + hour_particle_cells(hour_index,particle_index +j-1);
                        break;
                    end
                end

            end
        end


%FOR TESTING/REPORT PURPOSES ONLY:
```

```
%          figure(2);
%          imshow(binaryCy5);
%          figure(3);
%           imshow(brighterCy5);
%           figure(4);
%          imagesc(cell_matrix);
%          set(gca,'visible','off');
%          figure(5);
%          imshow(Cy5_image);
%          figure (6)
%          imagesc(bright_matrix);
%          set(gca,'visible','off');

        %set particle index to one past the current final particle:
        particle_index = size(particle_r, 1) + particle_index;
    end

    %obtain total particles for hour
    particles_for_hour(hour_index,1) = particle_index;
    hour = hour + 12;
end


%% create bar graph showing total cells and cells per particle and percent
empty particles
total_cells = zeros(9,1);
for i = 1:9
    total_cells(i,1) = sum(hour_particle_cells(i,:));
end

percent_empty = zeros(9,1);
for i = 1:9
    %count number of empty particles for each hour
    empty_counter = 0;
    for j = 1:particles_for_hour(i,1)
        if hour_particle_cells(i,j) == 0
            empty_counter = empty_counter + 1;
        end
    end

    %calculate percentage:
    percent_empty(i,1) = ( empty_counter / particles_for_hour(i,1) ) * 100;
end

%obtain total cells per non-empty particles
cells_per_particle = zeros(9,1);
for i = 1:9
    cells_per_particle(i,1) = total_cells(i,1) / (particles_for_hour(i,1) * (
(100 - percent_empty(i,1)) * 1/100 ));
end

%% Create bar graphs
figure(1);
hold on;
X = categorical({'Initial','12','24','36','48','60','72','84','96'});
X = reordercats(X, {'Initial','12','24','36','48','60','72','84','96'});
bar(X, total_cells);
```

```
xlabel('Hour')
ylabel('Total Cells')
hold off;


figure(2);
hold on;
bar(X, cells_per_particle);
xlabel('Hour')
ylabel('Cells Per Particle (Non-Empty)')
hold off;


figure(3);
hold on;
bar(X, percent_empty);
xlabel('Hour')
ylabel('Percent Empty')
ylim([0 100])
hold off;
```