Comet Assay

Lorenzo Torreggiani
Department of Information Engineering
Università Politecnica delle Marche
Ancona, Italy
lorenzo.torreggiani@outlook.com

***Abstract* - In this article will be shown the implementation of comet assay method.**

I. Introduction

Comet Assay, also called single cell gel electrophoresis (SCGE), is a sensitive and rapid technique for quantifying and analyzing DNA damage in individual cells.

As such, this is one of the techniques used in the area of cancer research for the evaluation of genotoxicity and effectiveness of chemoprevention.

The resulting image that is obtained resembles a "comet" with a distinct head and tail. The head is composed of intact DNA, while the tail consists of damaged (single-strand or double-strand breaks) or broken pieces of DNA.

II. Problem to Solve

The problem to solve is to extract the single comets from a rgb image and for each one of it, find:

- the histogram intensity;
- the diameter value of head of the comet;
- the length of tail of the comet.

III. Implementation

When the user loads a rgb image (Figure 1) the following steps are executed to achieve the objectives:

```
[fname,path] = uigetfile('*.png','Select an image');

fname = strcat(path,fname);

data = imread(fname);

figure, imshow(data);

title('Original image');
```

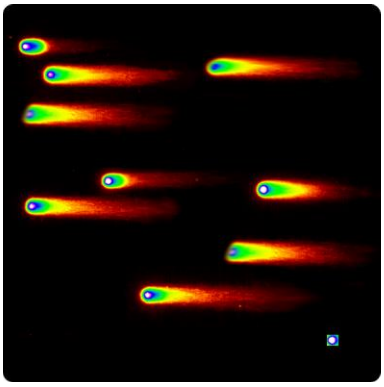

Figura 1 – Original image

- Convert the original image into grayscale image (Figure 2).

```
data_gray = rgb2gray(data);
figure, imshow(data_gray);
title('Grayscale image');
```

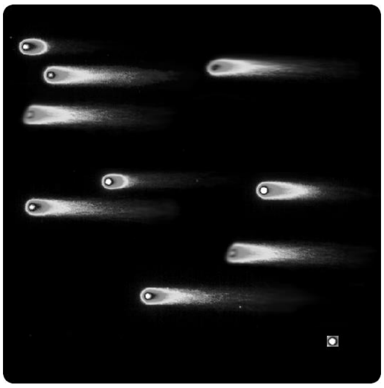

Figura 2 - Grayscale image

- Filter out noise using a median filter. It is a kind of smoothing filter that are used to prepare the image to next elaboration as segmentation. Within the median filter, each output pixel contains the median value in the m-by-n (3-by-3 in this case) neighborhood around the corresponding pixel in the input image.

```
data_gray = medfilt2(data_gray, [3 3]);
```

- Start the image segmentation through which, the image is divided into regions of interest, in this case the regions of interest are the different comets.

  The technique of segmentation used, is “thresholding”. It is a labelling operation on a gray scale image that distinguishes pixels of a higher intensity from pixels with a lower intensity value.

  The output of thresholding is a binary image (Figure 3) where the pixels inside the region of interest have a binary value of 1, while the pixels outside of the region of interest have a binary value of 0.

  Thresholding technique works well when the image contains objects on a uniform contrasting background.

  In details, at this point, the image is converted into a binary image using this parameters:

    - Method: adaptive, it depends from the coordinate x and y and it adapts the threshold value T for each pixel depending on local property of the image and gray scale value;
    - ForegroundPolarity: bright, it indicates that the foreground is brighter than background;
    - Sensitivity: 0.7, this factor for adaptive thresholding specified as a value in the range [0 1]. A high sensitivity value leads to thresholding more pixels as foreground, at the risk of including some background pixels.

```
data_bw = imbinarize(data_gray, 'adaptive',
'ForegroundPolarity', 'bright',
'Sensitivity', 0.7);

figure, imshow(data_bw), title('Binary image
after the segmentation');
```

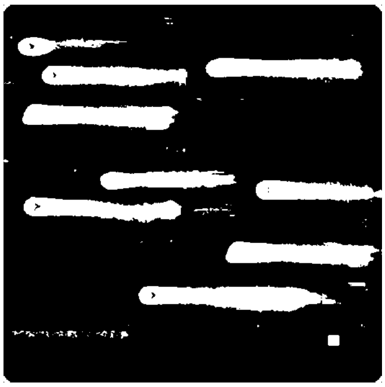

Figura 3 - Binary image after the segmentation

- In the image obtained from the previous point (Figure 4), it is removed all the areas less than 200 pixels.

```
data_bw = bwareaopen(data_bw, 200);

figure, imshow(data_bw), title('Binary
image');
```

Figura 4 - Binary image with no areas less than 200 pixel

- Start the Image Analysis finding connected component in the binary image.

```
bw = bwconncomp(data_bw, 8)
```

  “bw” is a structure that contains:

    - Type of connectivity: 8, it means that the pixel is centered in a neighborhood of 3x3 matrix;
    - Image size
    - NumObjects found: in this case 10;

- PixelIdList

- With the values of the structure “bw”, it is defined property of the different regions (comets, in this case) as Area, BoundingBox, CentroId, that will be useful to define the intensities values of the histograms, the diameters of the heads and the lengths of tails of comets.

```
stats = regionprops(bw, 'Area',
'BoundingBox', 'Centroid');
```

- At this stage, for each region found:
  - It is bounded each region in a rectangular box.
  - It is found the line where calculate the intensity value of each pixel. It is considered the central horizontal line (see Figure 4).
  - It is calculated the intensity values of the pixels along the segment and it is saved into a matrix but only the values that’s greater than zero.
  - It is calculated the max value of the intensity.
  - It is calculated the min value of the intensity considering min < max and min(n-1) > min.
  - It is calculated the value of the head of the comet considering the distance between the point of max and min.
  - It is calculated the value of the tail of the comet considering the distance between the point of max and the end of the segment where it is calculated the intensity values.
  - It is plotted: the comet, the values of the head and the tail of the comet, the image size, and the intensity values to obtain the intensity histogram.

```
batot = zeros(1);
bbtot = zeros(1,4);
bctot = zeros(1,2);
HH = zeros(1);
TT = zeros(1);

result = data_gray;

% This is a loop to:
% 1) Bound the objects in a rectangular box
% 2) Find line where calculate the
   intensity of each pixel
% 3) Find value of the head and the tail
for object = 1:length(stats)
```

```
ba = stats(object).Area;
bb = stats(object).BoundingBox;
bc = stats(object).Centroid;

% Save value of ba,bb,bc into different
matrix
batot(object,1) = ba;
for b = 1:4
    bbtot(object,b) = bb(1,b);
end
for c = 1:2
    bctot(object,c) = bc(1,c);
end

% 1) Bound the object in a rectangular
box
result = insertShape(result,
'Rectangle', bb, 'Color', 'green',
'LineWidth', 2);
result = insertText(result, [bb(1,1)-15
bb(1,2)], object, 'BoxOpacity', 1,
'FontSize', 10, 'BoxColor', 'green');

% 2) Find line where calculate the
intensity of each pixel
larg = bb(1,3);
lung = bb(1,4);
x1 = bb(1,1);
x2 = x1 + larg;
y1 = bb(1,2) + (lung/2);
y2 = y1;
x = [x1,x2];
y = [y1,y2];

% 3) Find value of the head and the tail
clear M2 M3

% improfile: get pixel-value along line
segment
[M2] = improfile(data_gray,x,y);

j = 1;
M3 = zeros(1);

% Get only the pixels with value > 0
for i = 1:length(M2)
   if M2(i,1) > 0
       M3(j,1) = M2(i,1);
       j=j+1;
   end;
end;
```

```
% Find max value in the matrix M3
max = 0;
max2 = 0;
m_max = 0;
for m = 1:length(M3)
    max2 = M3(m,1);
    if max2 >= max
        max = max2;
        m_max = m;
    end;
end;

% Find min value in the matrix M3
min = 0;
min2 = 0;
min3 = 0;
n_min = 0;
for n = 1:m_max
    min2 = M3(n,1);
    n1 = n-1;
    if n1 > 0
        min3 = M3(n1,1);
end;
    if and(min2 < max, min3 > min2)
        min = min2;
        n_min = n;
    end;
end;

head = m_max - n_min;
HH(object,1) = head;

tail = length(M3) - m_max;
TT(object,1) = tail;

% Print single comet and it's tail-value
figure, subplot(1,2,1);
crop = imcrop(result,bb);
imshow(crop);
title({['Comet ', num2str(object)],
['Image size: ', num2str(larg), 'x',
num2str(lung)], ['Diameter of the head:
', num2str(head), ' pixel'], ['Length of
the tail: ', num2str(tail), ' pixel']});

% Plot the histogram of the intensity
along the line-segments
subplot(1,2,2);
improfile(data_gray,x,y);
    title(['Histogram ',num2str(object)]);
    xlabel('Length of comet');
    ylabel('Color value');

end
```

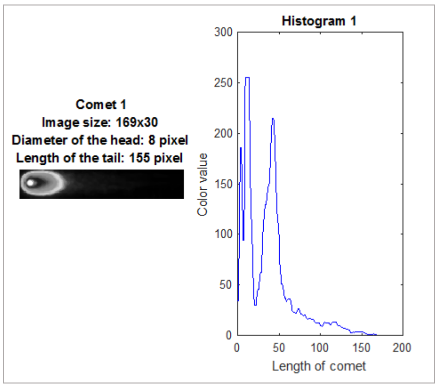

Figura 5 - Comet 1

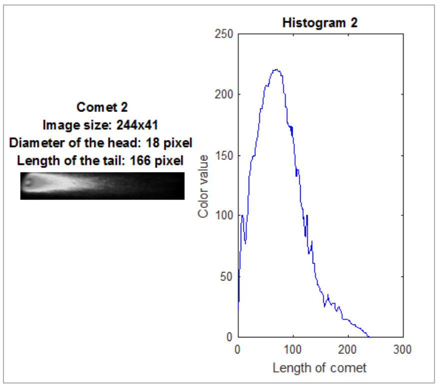

Figura 6 - Comet 2

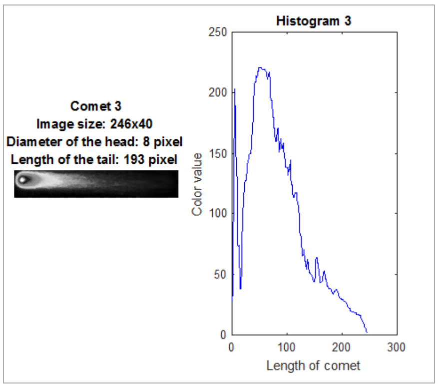

Figura 7 - Comet 3

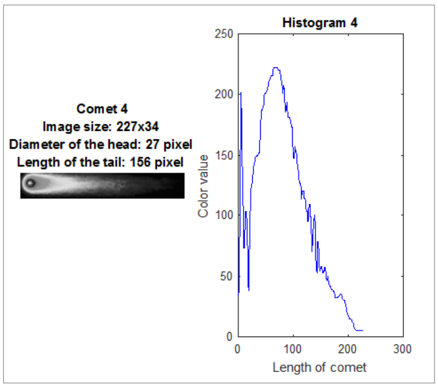

Figura 8 - Comet 4

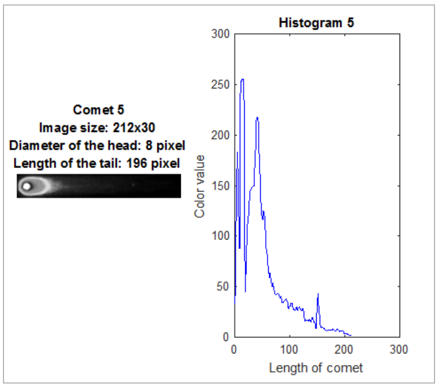

Figura 9 - Comet 5

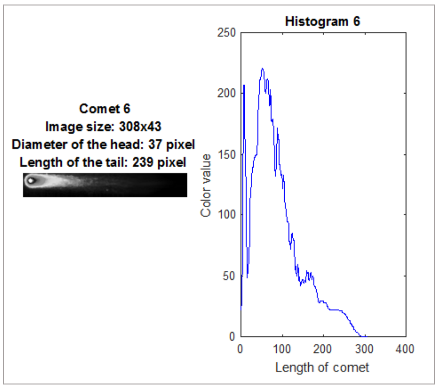

Figura 10 - Comet 6

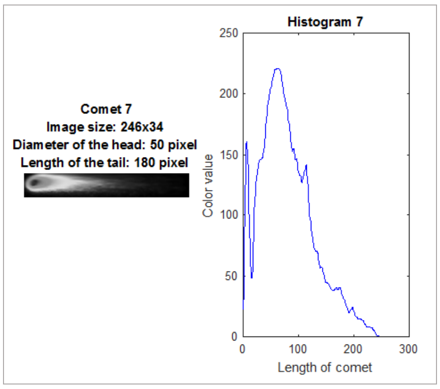

Figura 11 - Comet 7

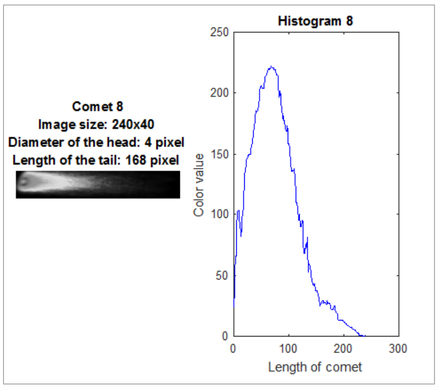

Figura 12 - Comet 8

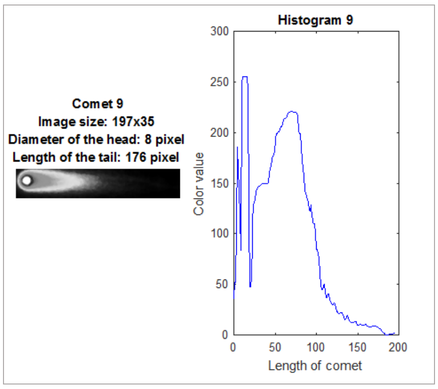

Figura 13 - Comet 9

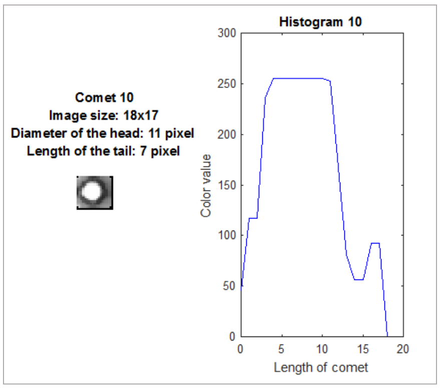

Figura 14 - Comet 10

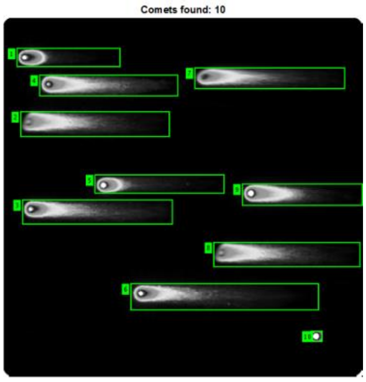

Figura 15 - Comets found

IV. CONCLUSION

This example shows how to create a simple program that extracts some useful information from an image.

In this test, I verified the values of the tails of the comets and I saw that, in the comets where, there is a substantial portion of green color on the right of the head of the comet, the tail is moved forward by a few pixels. This is demonstrated by the formula of "rgb to grayscale conversion" because the green color has a weight greater than the other colors (red and blue).

Gray = 0.21Red + 0.72Green + 0.07Blue

Same thing for the values of the heads of the comets because, at the point where start the tail, end the head, so if the tail is moved forward, also the head will be move forward.

In conclusion, the values of the heads and the tails are better when the max value of intensity histogram is on the red circle (see Figure 16).

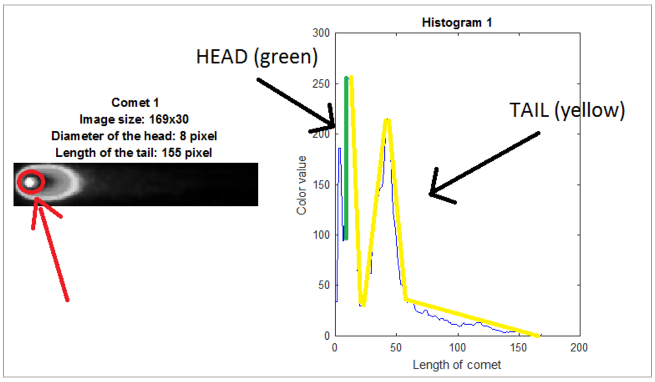

Figura 16

The next step of this project could be the possibility to adapt this code for a different types of input images.

REFERENCES

[1] http://www.sigmaaldrich.com/life-science/cell-biology/cancer-research/learning-center/cancer-research-protocols/comet-assay.html

[2] http://ispac.diet.uniroma1.it/scarpiniti/files/LabEMM/Less6.pdf

[3] http://www1.unipa.it/giovanni.petrucci/Disp/A_IntroMatlab_LEDIM.pdf

[4] https://it.mathworks.com/help/vision/examples/detecting-cars-using-gaussian-mixture-models.html

[5] https://it.mathworks.com/help/images/ref/imbinarize.htmlv