bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Quantitative Image Analysis of Tissue Properties: A MATLAB Tool for Measuring Morphology and Co-localization in 2D Images

Shira Landau[1,2], Erez Shor[3], Milica Radisic[1,2,4] and Shulamit Levenberg[3]

1. Institute of Biomedical Engineering, University of Toronto, Toronto, ON, Canada
2. Toronto General Hospital Research Institute, University Health Network, Toronto, ON, Canada
3. Department of Biomedical Engineering, Technion – Israel Institute of Technology, Haifa, Israel
4. Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, ON, Canada

Abstract

In recent years, the structural analysis of tissue elements has gained significant importance in biomedical research. Advancements in imaging technologies have created a pressing need to quantify tissue and cell properties accurately. This paper introduces a MATLAB-based analytical tool designed to measure a spectrum of properties from 2D images of tissues and cells. Our software efficiently computes parameters such as eccentricity, orientation, density, co-localization, size, and perimeter. The algorithm's precision in evaluating these characteristics has broad implications for enhancing the understanding of various biological processes and diseases. The code's flexibility allows for application across different tissue types and experimental conditions, providing researchers with a robust method for quantitative analysis. This advancement in computational image analysis represents a pivotal step towards more detailed and objective assessment in tissue engineering and cellular biology.

Introduction

The quantification of structural properties of tissue elements has emerged as a pivotal aspect of modern biomedical research, significantly aiding in the understanding of complex biological systems and disease pathologies. As imaging technology has advanced, the need for precise measurement of these properties has grown, necessitating the development of sophisticated computational tools. In this study, we introduce a MATLAB code capable of quantifying key properties of tissue or cell elements from 2D images.

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Quantitative image analysis has become indispensable in tissue engineering, developmental biology, and pathology. It allows researchers to characterize cellular morphology, identify phenotypic changes in response to treatments, and understand tissue architecture and its functional implications[1-3]. Traditional methods of analysis, often manual or semi-automated, are not only time-consuming but also prone to subjective bias, underscoring the necessity for automated, reliable, and reproducible computational techniques[1].

The MATLAB code presented in this paper addresses these challenges by automating the measurement of eccentricity, which quantifies the deviation of a cell's shape from a perfect circle[4]; orientation, which determines the angle of the major axis of the cell's ellipse to a reference axis; density, which assesses the number of cells per unit area; co-localization, which evaluates the spatial overlap of different cellular markers; size, which measures the area covered by a cell or group of cells; and perimeter, which calculates the boundary length of the cells.

This comprehensive approach to analyzing tissue and cell images leverages MATLAB's robust computational capabilities, allowing for the processing of large datasets with precision and efficiency. The code is designed to be user-friendly, requiring minimal user input and providing clear, interpretable results that can be further analyzed statistically. It is adaptable to a wide range of tissues and cells, making it an invaluable tool for multiple research domains.

Moreover, this tool's introduction into the scientific community has the potential to standardize quantitative analysis methodologies across laboratories, fostering consistent and comparative studies. By automating complex analysis, it enables researchers to focus on experimental design and interpretation of results, ultimately accelerating the pace of discovery.

In summary, the MATLAB code developed for this study stands as a significant contribution to the field of quantitative image analysis. Its implementation will facilitate a deeper understanding of the structural properties of tissues and cells, paving the way for breakthroughs in biomedical research and diagnostic procedures.

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Procedure

Image Collection

In this study, the MATLAB code was developed to analyze 2D images of tissues and cells of RGB images. These images can be either JPEG or TIFF formats at any resolution or magnification. When comparing between groups or treatments, images should be captured using uniform acquisition parameters, such as laser gain and exposure settings, as well as maintaining consistent magnification and image size. The use of a scale bar should be avoided.

If images exhibit a biased signal due to different staining protocols or tissue thickness, it could skew the results. This issue can be addressed by applying contrast enhancement to the dimmer images.

Matlab requirement

Any version of MATLAB from 2013 onwards can be used, and the Image Processing Toolbox should also be installed.

MATLAB Code Description

1. Reading the image

```
clear all;
close all;
close all;
% reading the image
[filename,pathname]=uigetfile('*.*','Select Image');
im = imread(strcat(pathname,filename),'tiff'); %change to jpg if needed
```

2. Selecting a region of interest (ROI):

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

If only part of the image is analyzed and a manual selection is required, this part of the code would be used:

```
ROI = roipoly(im); %for manual selection use this
```

If the whole image is being analyzed

```
ROI=1;
```

3. Dividing the image into 3 RGB channels:

```
%% Dividing the image into 3 RGB channels:
im_red=double(im(:,:,1)).*ROI/255; % 1=red 2=green 3=blue
im_green=double(im(:,:,2)).*ROI/255;% 1=red 2=green 3=blue
im_blue=double(im(:,:,3)).*ROI/255;% 1=red 2=green 3=blue
```

4. Converting the images into binary images using a defined threshold

```
%% Converting the images into binary images using a defined threshold
% For the red channel:
thresh_red=0.3; % Adjust to your satisfaction
imBW_red=im2bw(im_red,thresh_red);
figure;
imagesc(imBW_red);
% For the green channel:
thresh_green=0.1; % Adjust to your satisfaction
imBW_green=im2bw(im_green,thresh_green);
figure;
imagesc(imBW_green);
% For the blue channel:
thresh_blue=0.3; % Adjust to your satisfaction
imBW_blue=im2bw(im_blue,thresh_blue);
figure;
imagesc(imBW_blue);
```

For co-localization analysis:

```
%% for co-localization analysis
imBW_green_red=imBW_green.*imBW_red; %co localization channels could be changed according to the required analysis
figure;
imagesc(imBW_green_red);
```

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Note: Images are displayed at this point for you to fine-tune the thresholding parameters until satisfied.

1. Measure Image properties with the regioprops function:

```
%% Measure Image properties with the regioprops function:
S_red= regionprops(imBW_red,'All');
S_green= regionprops(imBW_green,'All');
S_blue= regionprops(imBW_blue,'All');
```

2. Image properties of the red channel

```
%% Image properties of the red channel
N=1;
for i=1:size(S_red,1)
   if S_red(i).Area>10 %change size according to background noise
      Eccentricity_red(i)=S_red(i).Eccentricity;
      orient_red(i)=S_red(i).Orientation;
      complexity_red(i)=1/(S_red(i).Area/(S_red(i).Perimeter^2)*4*pi);
       length_red(i)=S_red(i).MajorAxisLength;
        Perimeter_red(i)=S_red(i).Perimeter;
      N=N+1;
   end;
end;
```

3. Image properties of the green channel

```
%% Image properties of the green channel
N=1;
for i=1:size(S_green,1)
   if S_green(i).Area>10
      orient_green(i)=S_green(i).Orientation;
      Eccentricity_green(i)=S_green(i).Eccentricity;
      complexity_green(i)=1/(S_green(i).Area/(S_green(i).Perimeter^2)*4*pi);
       length_green(i)=S_green(i).MajorAxisLength;
       Perimeter_green(i)=S_green(i).Perimeter;
      N=N+1;
   end;
end;
```

4. Image properties of the blue channel

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

```
%% Image properties of the blue channel
N=1; nuclei_num=0;
for i=1:size(S_blue,1)
   if S_blue(i).Area>10
      orient_blue(i)=S_blue(i).Orientation;
      Eccentricity_blue(i)=S_blue(i).Eccentricity;
      complexity_blue(i)=1/(S_blue(i).Area/(S_blue(i).Perimeter^2)*4*pi);
       length_blue(i)=S_blue(i).MajorAxisLength;
       Perimeter_blue(i)=S_blue(i).Perimeter;
        Blue_size(i)=S_blue(i).Area; %nucleus size
        nuclei_num=nuclei_num+1; %counting number of nuclei
      N=N+1;
   end;
end;
```

Note 1: change the area size filter according to your noise particle size.

5. Averages calculation for all parameters and channels

```
%% averages calculation for all parameters and channels
avg_blue_size=mean(Blue_size);

density_red=sum(sum(imBW_red))/size(im,1);
density_green=sum(sum(imBW_green))/size(im,1);
density_blue=sum(sum(imBW_blue))/size(im,1);
density_green_red=sum(sum(imBW_green_red))/size(im,1);
Nor_density_red=density_red/density_blue;
Nor_density_green=density_green/density_blue;


% Eccentricity analysis
Eccentricity0_25_red=100*sum((Eccentricity_red>0)&(Eccentricity_red<0.25))/size(Eccentric
ity_red,2); %this value will be of the most round elements
Eccentricity25_50_red=100*sum((Eccentricity_red>0.25)&(Eccentricity_red<0.5))/size(Eccen
tricity_red,2);
Eccentricity50_75_red=100*sum((Eccentricity_red>0.5)&(Eccentricity_red<0.75))/size(Eccen
tricity_red,2);
Eccentricity75_80_red=100*sum((Eccentricity_red>0.75)&(Eccentricity_red<0.8))/size(Eccen
tricity_red,2);
Eccentricity80_85_red=100*sum((Eccentricity_red>0.8)&(Eccentricity_red<0.85))/size(Eccen
tricity_red,2);
Eccentricity85_90_red=100*sum((Eccentricity_red>0.85)&(Eccentricity_red<0.9))/size(Eccen
tricity_red,2);
```

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

```
Eccentricity90_95_red=100*sum((Eccentricity_red>0.9)&(Eccentricity_red<0.95))/size(Eccentricity_red,2);
Eccentricity95_100_red=100*sum((Eccentricity_red>0.95)&(Eccentricity_red<1))/size(Eccentricity_red,2); %this value will be of the most elongated elements

%orientation analysis
elements0=sum(abs(orient_red)<30);
elements90=sum(abs(orient_red)>60);
elementsOther=size(orient_red,2)-elements90-elements0;

%orientation variance
VarOri_red=var(abs(orient_red));
```

6. Writing the results into a CSV file:

```
%% Writing the results into a csv file
fid = fopen('results.csv', 'a+');%name of excel file
fprintf(fid,'file name,Averaged nuclei size(pix),Normalized Red channel density,Normalized green channel density,Elements of eccentricity 0.95-1 in red,Orientation variance in red, \n');

fprintf(fid,'%s, %f, %f, %f, %f, %f, \n',filename,avg_blue_size,Nor_density_red,Nor_density_green,Eccentricity95_100_red,VarOri_red);
fclose(fid);
```

If multiple images are used, the "*Image Batch Processor*" APP is recommended. Using this adjusted function:

```
function results = ImageAnalysis(im)
ROI=1;
%ROI = roipoly(im); %for manual selection use this

% Dividing the image into 3 RGB channels:
im_red=double(im(:,:,1)).*ROI/255; % 1=red 2=green 3=blue
im_green=double(im(:,:,2)).*ROI/255;% 1=red 2=green 3=blue
im_blue=double(im(:,:,3)).*ROI/255;% 1=red 2=green 3=blue

% Converting the images into binary images using a defined threshold
% For the red channel:
thresh_red=0.3; % Adjust to your satisfaction
imBW_red=im2bw(im_red,thresh_red);
```

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

```
figure;
imagesc(imBW_red);
% For the green channel:
thresh_green=0.1; % Adjust to your satisfaction
imBW_green=im2bw(im_green,thresh_green);
figure;
imagesc(imBW_green);
% For the blue channel:
thresh_blue=0.3; % Adjust to your satisfaction
imBW_blue=im2bw(im_blue,thresh_blue);
figure;
imagesc(imBW_blue);
% for co-localization analysis
imBW_green_red=imBW_green.*imBW_red; %co localization channels could be changed according to the required analysis
figure;
imagesc(imBW_green_red);

% Measure Image proporties with the regioprops function:
 S_red= regionprops(imBW_red,'All');
 S_green= regionprops(imBW_green,'All');
 S_blue= regionprops(imBW_blue,'All');

% Image proporties of the red channel
N=1;
for i=1:size(S_red,1)
   if S_red(i).Area>10 %change size according to backround noise
      Eccentricity_red(i)=S_red(i).Eccentricity;
      orient_red(i)=S_red(i).Orientation;
      complexity_red(i)=1/(S_red(i).Area/(S_red(i).Perimeter^2)*4*pi);
       length_red(i)=S_red(i).MajorAxisLength;
        Perimeter_red(i)=S_red(i).Perimeter;
      N=N+1;
   end;
end;
% Image proporties of the green channel
N=1;
for i=1:size(S_green,1)
   if S_green(i).Area>10
      orient_green(i)=S_green(i).Orientation;
      Eccentricity_green(i)=S_green(i).Eccentricity;
      complexity_green(i)=1/(S_green(i).Area/(S_green(i).Perimeter^2)*4*pi);
       length_green(i)=S_green(i).MajorAxisLength;
       Perimeter_green(i)=S_green(i).Perimeter;
      N=N+1;
   end;
```

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

```
end;

% Image proporties of the blue channel
N=1; nuclei_num=0;
for i=1:size(S_blue,1)
   if S_blue(i).Area>10
      orient_blue(i)=S_blue(i).Orientation;
      Eccentricity_blue(i)=S_blue(i).Eccentricity;
      complexity_blue(i)=1/(S_blue(i).Area/(S_blue(i).Perimeter^2)*4*pi);
       length_blue(i)=S_blue(i).MajorAxisLength;
       Perimeter_blue(i)=S_blue(i).Perimeter;
        Blue_size(i)=S_blue(i).Area; %nucleus size
        nuclei_num=nuclei_num+1; %counting number of nuclei
      N=N+1;
   end;
end;

% average caculation for all parameters and channels
avg_blue_size=mean(Blue_size);

density_red=sum(sum(imBW_red))/size(im,1);
density_green=sum(sum(imBW_green))/size(im,1);
density_blue=sum(sum(imBW_blue))/size(im,1);
density_green_red=sum(sum(imBW_green_red))/size(im,1);
Nor_density_red=density_red/density_blue;
Nor_density_green=density_green/density_blue;


% Eccentricty analysis
Eccentricity0_25_red=100*sum((Eccentricity_red>0)&(Eccentricity_red<0.25))/size(Eccentric
ity_red,2); %this value will be of the most round elements
Eccentricity25_50_red=100*sum((Eccentricity_red>0.25)&(Eccentricity_red<0.5))/size(Eccen
tricity_red,2);
Eccentricity50_75_red=100*sum((Eccentricity_red>0.5)&(Eccentricity_red<0.75))/size(Eccen
tricity_red,2);
Eccentricity75_80_red=100*sum((Eccentricity_red>0.75)&(Eccentricity_red<0.8))/size(Eccen
tricity_red,2);
Eccentricity80_85_red=100*sum((Eccentricity_red>0.8)&(Eccentricity_red<0.85))/size(Eccen
tricity_red,2);
Eccentricity85_90_red=100*sum((Eccentricity_red>0.85)&(Eccentricity_red<0.9))/size(Eccen
tricity_red,2);
Eccentricity90_95_red=100*sum((Eccentricity_red>0.9)&(Eccentricity_red<0.95))/size(Eccen
tricity_red,2);
Eccentricity95_100_red=100*sum((Eccentricity_red>0.95)&(Eccentricity_red<1))/size(Eccent
ricity_red,2); %this value will be of the most elongated elements
```

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

```
%orientation analysis
elements0=sum(abs(orient_red)<30);
elements90=sum(abs(orient_red)>60);
elementsOther=size(orient_red,2)-elements90-elements0;

%orientation variance
VarOri_red=var(abs(orient_red));

% Writing the results into a csv file
fid = fopen('results2.csv', 'a+');%name of excel file
fprintf(fid,'Averaged nuclei size(pix),Normalized Red channel density,Normalized green channel density,Elements of eccentricity 0.95-1 in red,Orientation variance in red, \n');

fprintf(fid,'%f, %f, %f, %f, %f, \n',avg_blue_size,Nor_density_red,Nor_density_green,Eccentricity95_100_red,VarOri_red);
fclose(fid);
end
```

Example

This example outlines the measurement of elements within an in vitro tissue composed of GFP-expressing endothelial cells, dental pulp stem cells (DPSCs), and cardiomyocytes embedded in a gel and stained with the cardiac marker MLC2V.

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

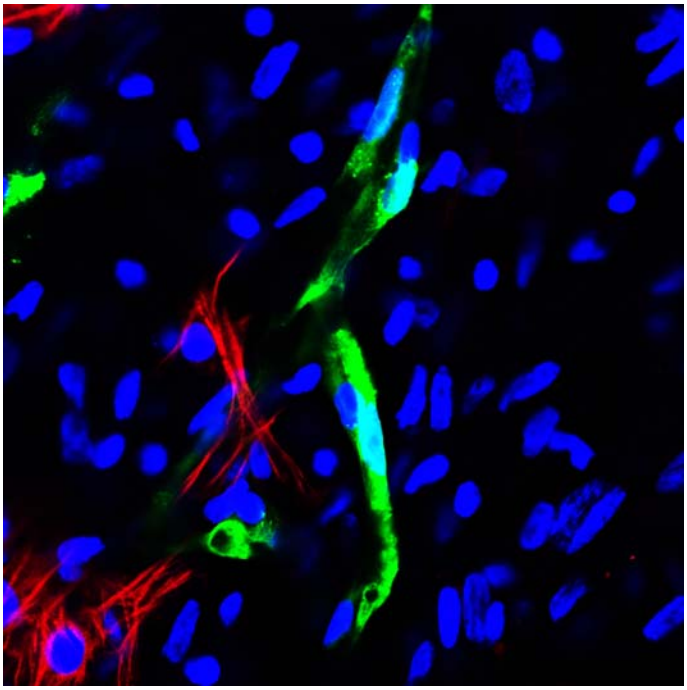


Figure 1: in vitro culture featuring GFP-expressing endothelial cells (green), cardiomyocytes, and DPSCs, stained with DAPI (blue) and MLC2V (red).

Subsequently, the image is segmented into three binary images representing each fluorescent channel (**Figure 1**).

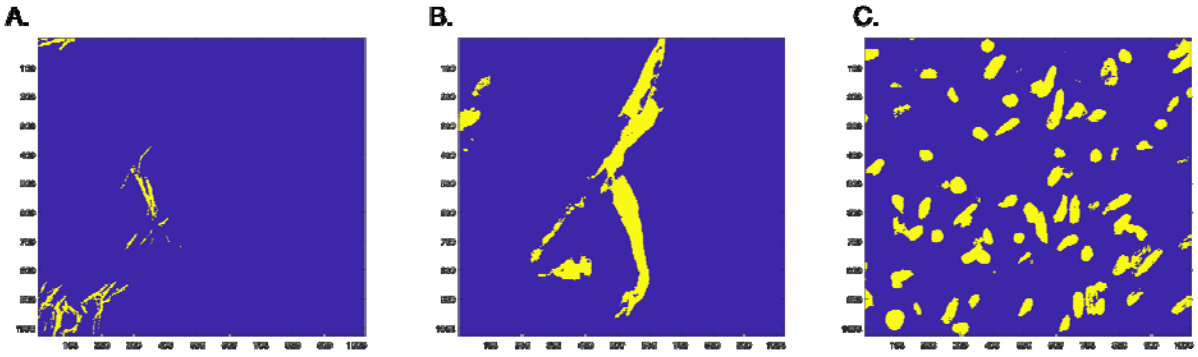
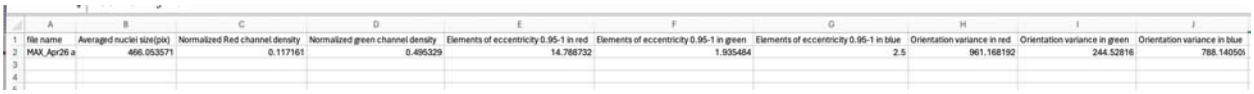


Figure 2: Binary images of the three channels following appropriate thresholding: **A.** red, **B.** green, and **C.** blue channels.

These images are processed using the *regionprops* function to extract the necessary measurements, which are then cataloged in an Excel spreadsheet, organizing each measurement into separate columns and each sample into distinct rows (**Figure 2**).

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



	A	B	C	D	E	F	G	H	I	J
1	file name	Averaged nuclei size(pix)	Normalized Red channel density	Normalized green channel density	Elements of eccentricity 0.95-1 in red	Elements of eccentricity 0.95-1 in green	Elements of eccentricity 0.95-1 in blue	Orientation variance in red	Orientation variance in green	Orientation variance in blue
2	MAX_Apr26 a	466.053571	0.117161	0.495329	14.788732	1.935484	2.5	961.168192	244.52816	788.140505
3										
4										

Figure 3: the results table from the image analysis. In this example, the following parameters were measured: average nucleus size, red channel density normalized to nuclei, green channel density normalized to nuclei, the percentage of elements with an eccentricity value between 0.95 and 1 in the red, green, and blue channels, and the variance in orientation across the red, green, and blue channels.

The results indicate that the normalized density of the green channel exceeds that of the red channel, as expected. Additionally, the red channel, as anticipated, contained the highest percentage of elements with an eccentricity value between 0.95 and 1. In terms of orientation, the green channel exhibited the lowest variance, indicating the most aligned features (F**igure 3**).

Summary

The study introduces a MATLAB-based analytical tool capable of accurately measuring a range of properties from 2D images of tissues and cells, such as eccentricity, orientation, density, and size. Enhanced by a batch processing feature, the software can swiftly analyze hundreds of images, offering a significant increase in efficiency and making it a robust and versatile solution for comprehensive quantitative analysis in tissue engineering and cellular biology.

References

1. Wiesmann, V., Franz, D., Held, C., Münzenmayer, C., Palmisano, R., and Wittenberg, T. (2015). Review of free software tools for image analysis of fluorescence cell micrographs. Journal of microscopy *257*, 39-53.
2. Kaifosh, P., Zaremba, J.D., Danielson, N.B., and Losonczy, A. (2014). SIMA: Python software for analysis of dynamic fluorescence imaging data. Frontiers in neuroinformatics *8*, 80.

bioRxiv preprint doi: https://doi.org/10.1101/2024.04.03.587971; this version posted April 5, 2024. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

3. Hamilton, N.A. (2012). Open source tools for fluorescent imaging. Methods in enzymology *504*, 393-417.
4. Hage, P., and Harary, F. (1995). Eccentricity and centrality in networks. Social networks *17*, 57-63.