Marjolein van Tol
January 2025

Bachelor thesis: Bio-informatica

MANUAL FIBRIL ANALYSIS TOOL

TABLE OF CONTENTS

About the Tool ***1***

Suitable Images **1**

The Methods ***2***

1) Fibril Analysis **2**

2) Sacromere Analysis Tracing **2**

3) Sacromere Analysis FFT **3**

Calculating Bandwidth ***4***

Instructions for the tool ***6***

1) Preproccesing the images 6
2) Using the Tool 9
3) The results 10

References ***11***

ABOUT THE TOOL

The fibril analysis tool is a graphical application designed for automated analysis of myofibril structures. It focuses on the detection and measurement of structural features such as fibrils, sarcomeres, and Z-disks in microscopy images. The users can choose from three main methods:

- Fibril Analysis,
- Sarcomere Analysis Tracing
- Sarcomere Analysis FFT

Each analysis option includes configurable parameters that users can adjust according to their dataset's requirements, with default values available for guidance.
Since this is an early version, some steps require manual input from the user. This manual provides detailed instructions on using the tool correctly and explains how each algorithm functions.

SUITABLE IMAGES

The images suitable for this program should have clear z-disk labeling. The image should have one channel in 16-bit format. When there is too much noise in the images, a clear analysis will be more difficult. Always visually assess the quality of the images and results when analyzing images.

THE METHODS

As noted in the introduction, each algorithm will be briefly described to give users a clearer understanding of how they work. In the tutorial, a more detailed explanation of the tool's usage is also provided to help improve the user's understanding.

1) FIBRIL ANALYSIS

The first algorithm was created by Linde Schoenmaker in 2021 and serves as a full analysis of the fibril structures in the image. It identifies fibrils in each image, calculating different metrics [2].

Image 1: Visual representation of the fibril tracing done by the algorithm. First image is a original control hart image of the top left of the sample. Second image shows a binary mask of the original image. Right image shows the fibril tracing.

In Image 1, we see a visual representation of the fibril tracing generated by the algorithm. First, a binary mask of the original image is created using ImageJ (instructions provided in the tutorial).

For the initial image, Z-disks are labeled to highlight their locations, serving as reference points for the algorithm. The binary mask retrieves these Z-disks and establishes a coordinate system for further calculations. After preliminary image cleaning, the algorithm identifies neighboring Z-disks (forming sarcomeres) and connects them, repeating this process to trace entire fibrils.

Once tracing is complete, the algorithm calculates key metrics, including:
- Number of fibrils
- Average length and width of the fibrils (with standard deviations)
- Organized orientation (using a Chi-square test p-value for distribution)
- Full orientation distribution (| means vertical, - horizontal, / and \ diagonal)

2) SACROMERE ANALYSIS TRACING

The sarcomere analysis is one of the two methods programmed by Tijmen van Wel in 2023 [3]. The algorithm identifies sarcomere structures by using the angles between z-lines. In image 2 we can see a visual representation of the tracing done by the algorithm.

We begin by creating a binary mask of the original image to extract the z-disk structures. Each z-disk is then identified, with smaller disks being filtered out. Next, based on the orientation of the remaining

z-disks, a perpendicular line is drawn between pairs of disks that share the same orientation. This process creates the sarcomere structures between the z-disks.

Image 2: Visual representation of the sarcomere tracing performed by the algorithm. The first image is an original control heart image from the middle right of the sample. The second image shows a binary mask of the original image with skeletonization. The rightmost image displays the z-disk tracing (in red) and the sarcomere structures (in blue).

Once tracing is complete, the algorithm calculates the following metrics:
- Average sarcomere length
- Amount of traced z-disks
- Number of created lines between z-disks (amount of sarcomere structures that are traced)
- Ratio between number of z-lines and connected lines

3) SACROMERE ANALYSIS FFT

The last method uses the FFT method for calculating sarcomere lengths. The code used for this method was made by Tijmen van Wel in 2023 [3].

We apply a 2D Fast Fourier Transform (2D-FFT) to the image to analyze its frequency content. In the original image, each pixel represents a specific brightness or intensity, reflecting spatial information. The 2D-FFT converts this spatial information into the frequency domain, where each pixel in the transformed image now corresponds to a frequency component. Bright spots indicate repeating patterns, like sarcomere spacing.

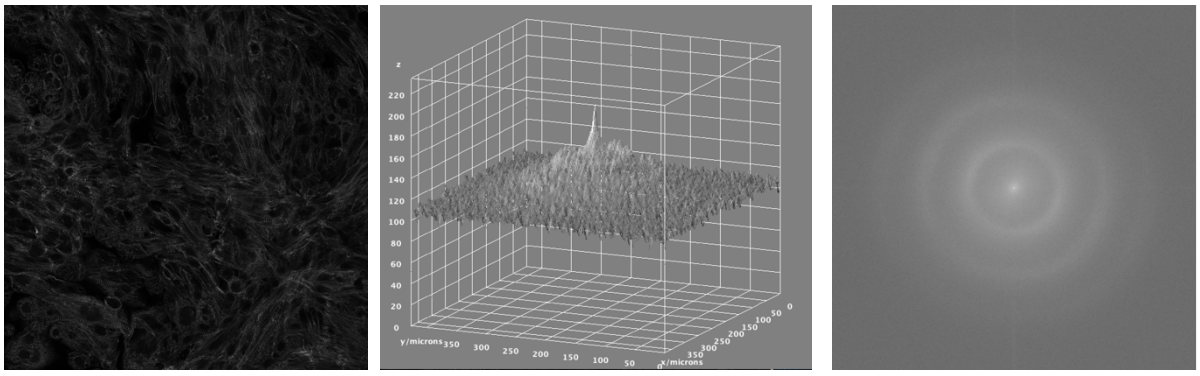


Image 3: Visual representation of the 2D-FFT transformation of an image. The first image is an original control heart image from the middle right of the sample. The second image shows the 3D frequency domain of the spatial information. The right image is a 2D conversion of the 3D representation.

To reduce noise, we set frequency bounds based on typical sarcomere lengths, ensuring we focus on meaningful frequencies (seen in image 4). This approach also helps retrieve binary masks in the previous methods. Finally, we calculate the total brightness and the number of pixels at each distance, creating an average brightness distribution. In Image 4, the peak of this distribution represents the average sarcomere length, which we then convert to micrometers (µm).

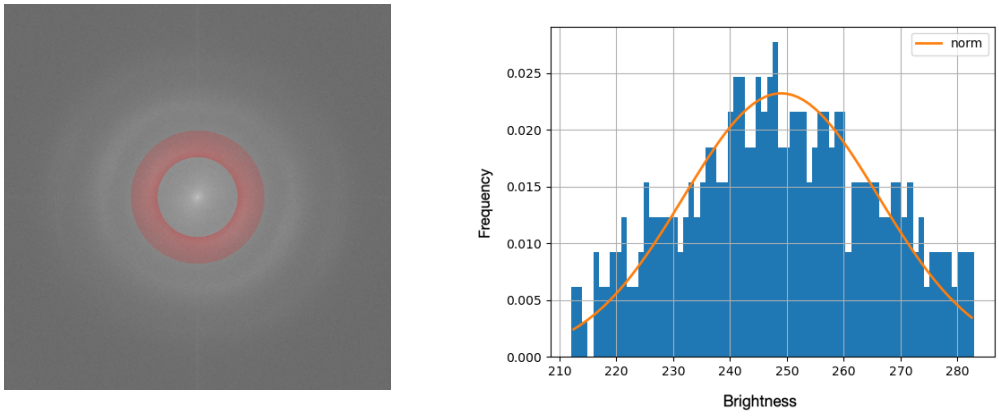


Image 4: Right images shows the 2D-FFT with the frequency bound (red space). Left image shows the brightness distribution within the frequency bound. The peak represents the average sarcomere length before converging back to micrometers.

The algorithm returns the following information:

- Average sarcomere length

CALCULATING BANDWIDTH

In the first two methods, we use the bandwidth to create a frequency bound and set a threshold to remove noise and segment the z-lines. The sarcomere lengths are typically 1.6–1.8 µm in hPSC-CMs and 2.2 µm in mature human adult CMs [2]. Therefore, the average sarcomere length is set at 1.6 µm, with a bandpass filter range of 1.2–2.0 µm. This means that only sarcomeres within this size range will be considered.
The conversion of the bandpass filter to the frequency domain is calculated using the image's dimensions, the pixel-to-micrometer ratio, and a formula to determine frequency boundaries [1]:

$$Frequency = \frac{N}{R * SL}$$

N: Closest power of two with respect to the max(width, height) of the image
R: The pixel-to-micrometer ratio
S: Upper frequency bound
L: Lower frequency bound

This frequency is then used in the binary mask conversion for the first two methods. It is also utilized in the 2D-FFT conversion, as seen in that section. However, in this method, the calculations are automated within the script. For more information see paper [2].

Image properties

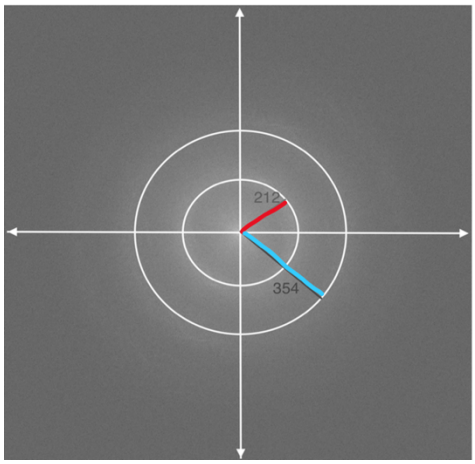
$Image\ dimensions$: $2048x2048$

$Ratio$: 4.8177

S/L: $1.2 - 2.0\ \mu m$

Calculations

$$N = \max(height, width)$$
$$N = \max(2048x2048)$$
$$N = 2048$$



$$Frequency = \frac{N}{R * SL}$$

$$Frequency_{upper\ bound} = \frac{2048}{4.8177 * 1.2} = 354.25$$
$$Frequency_{lower\ bound} = \frac{2048}{4.8177 * 2.0} = 212.55$$

Adjust script

$$Lower\ coordinate = 1024 - Frequency_{lower\ bound}$$
$$Lower\ Fz = Frequency_{lower\ bound}\ x\ 2$$

$$Higher\ coordinate = 1024 - Frequency_{higher\ bound}$$
$$Higher\ Fz = Frequency_{higher\ bound}\ x\ 2$$

cardiotox_macro.ijm

Run Install Macro

```
run("8-bit");
run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");
run("Sharpen");
run("FFT");
setColor(0);
makeOval(Lower coordinate, Lower coordinate, Lower Fz, Lower Fz)
fill();
makeOval(Higher coordinate, Higher coordinate, Higher Fz, Higher Fz)
run("Make Inverse");
fill();
run("Inverse FFT");
a=getTitle();
selectImage(a);
run("Enhance Contrast...", "saturated=0.2 normalize");
setThreshold(80, 255);
run("Convert to Mask");
run("Analyze Particles...", "summarize");
//saveAs("Tiff", binaryDir + "/"+ list[i]);
```

cardiotox_macro.ijm

Run Install Macro

```
run("8-bit");
run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");
run("Sharpen");
run("FFT");
setColor(0);
makeOval(811, 811, 425, 425);
fill();
makeOval(670, 670, 708, 708);
run("Make Inverse");
fill();
run("Inverse FFT");
a=getTitle();
selectImage(a);
run("Enhance Contrast...", "saturated=0.2 normalize");
setThreshold(80, 255);
run("Convert to Mask");
run("Analyze Particles...", "summarize");
//saveAs("Tiff", binaryDir + "/"+ list[i]);
```

INSTRUCTIONS FOR THE TOOL

Before using any of the methods, it's important to preprocess the images to ensure they are in the right format and quality for analysis. As previously mentioned, we will use ImageJ, a powerful and widely used image processing software, to handle this preprocessing.

1) PREPROCCESING THE IMAGES

- First, you will need to download ImageJ. You can use the following link to the ImageJ website: https://imagej.net/ij/download.html

- Once you have successfully downloaded ImageJ, you can open the program.

  Different algorithms will require different preprocessing steps:

PREPROCESSING **BINARY MASK** (FOR FIBRIL AND SARCOMERE TRACING ANALYSIS)

1. Updating the ImageJ *cardiotox_macro.ijm file*
   - With the main folder, a file is provided: *cardiotox_macro.ijm* (see image 5). This file contains the code that we will need to process the images.
   - Open the *cardiotox_macro.ijm* file by dragging it into the ImageJ menu window.
   - Now you will need to change the bandwidth filter based on your image properties:

     In section Method > Calculation Bandwidth, you can find a more in-depth explanation and an example on how to calculate the bandwidth based on your image properties.
     - Calculate the frequency bound with your image properties.
     - Change the *makeOval ()* lines with your new bound:

cardiotox_macro.ijm

Run Install Macro

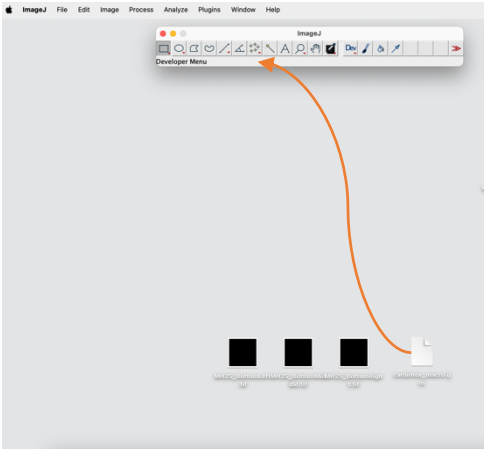
```
run("8-bit");
run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");
run("Sharpen");
run("FFT");
setColor(0);
makeOval (Lower coordinate, Lower coordinate, Lower Fz, Lower Fz)
fill();
makeOval (Higher coordinate, Higher coordinate, Higher Fz, Higher Fz)
run("Make Inverse");
fill();
run("Inverse FFT");
a=getTitle();
selectImage(a);
run("Enhance Contrast...", "saturated=0.2 normalize");
setThreshold(80, 255);
run("Convert to Mask");
run("Analyze Particles...", "summarize");
//saveAs("Tiff", binaryDir + "/"+ list[i]);
```

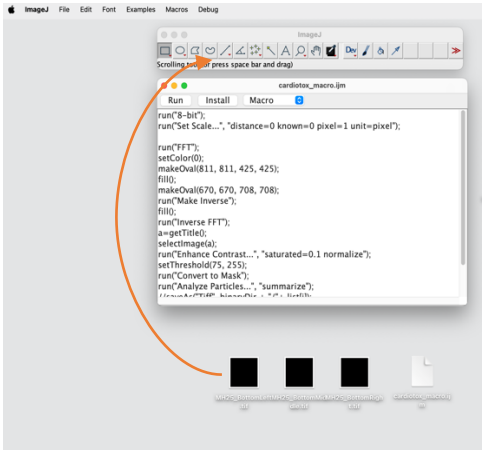
$$Lower\ coordinate = 1024 - Frequency_{lower\ bound}$$
$$Lower\ Fz = Frequency_{lower\ bound}\ x\ 2$$

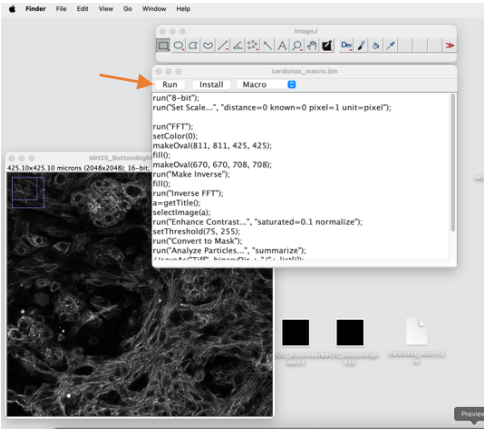
$$Higher\ coordinate = 1024 - Frequency_{higher\ bound}$$
$$Higher\ Fz = Frequency_{higher\ bound}\ x\ 2$$

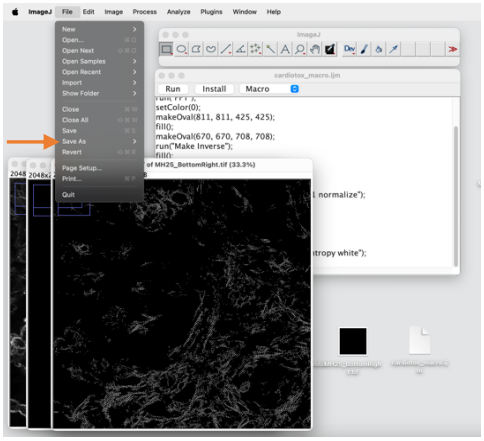
2. Creating Binary Masks

- Next, open the TIFF images that you want to analyze by also dragging them into the ImageJ menu window.
- For each image, run the *cardiotox_macro.ijm* code by clicking the "Run" button in the ImageJ menu. Multiple images will show on your screen. Only the final image with the binary mask needs to be saved.
- Save each processed image as a ".tiff " file and store them in a separate folder for later use.
- Now, you will have created binary masks for each image, which can then be used for the Fibril and Sarcomere methods.



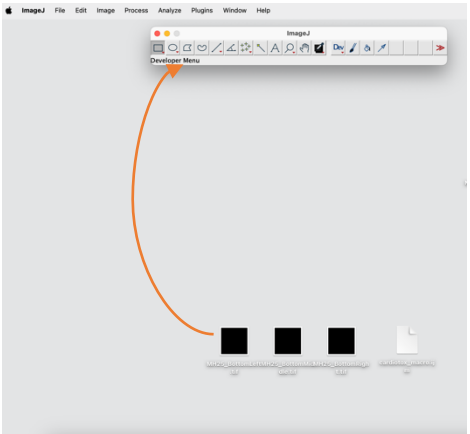


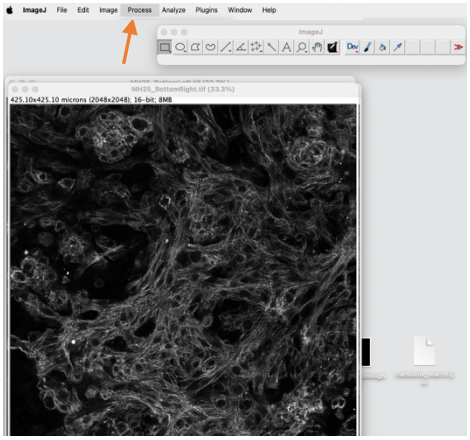


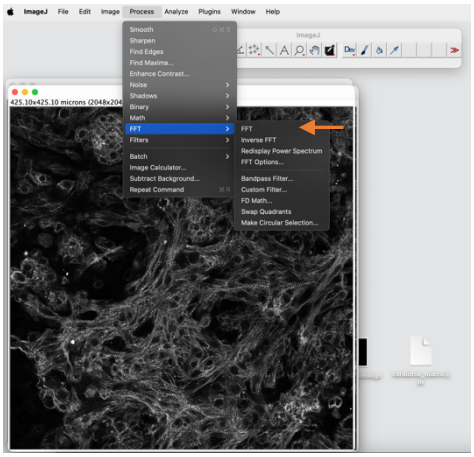


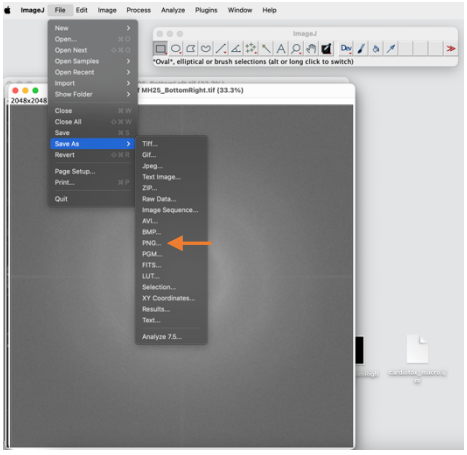
PREPROCESSING **2D-FFT** FOR THE SARCOMERE FFT ANALYSIS

- Open the TIFF images that you want to analyze by dragging them into the ImageJ menu window.
- Run the FFT Conversion by selecting “Process > FFT > FFT” in the top menu.
- Make sure to save the 2D-FFT conversion of the images as **“.png ”** files and store them in a separate folder for later use.
- Now, you will have created the 2D-FFT conversions for each image, which can then be used for the 2D-FFT method.









2) USING THE TOOL

Now that the images are preprocessed we can start using the Tool.

- Open the Tool by going to dist > Tool > Tool.exe (see image 5)
  It can take some time for the tool to open

- First we have to upload the files that we want to analyze. Click the “upload” button and select the files you want to analyze. (Make sure that the images that you upload are compatible with the algorithm that you want to run)
  - You can delete previous images with the delete button.

- For each algorithm it is important that you enter in the right ratio of the images in the entry field.

- Select the algorithm you want to use in the dropdown at the top of the window.

- For each algorithm there are different parameters that can be adjusted. Reset the parameters to their default value by pressing the “reset” button.

- Run the algorithm by pressing the “run” button at the bottom of the window. You will see which file is being processed.
  The "sarcomere analysis tracing" method takes quite a long time to run compared to other methods.

- Once the algorithm is done, a csv file will be opened that shows the results.
  - In the folder where you can find the tool, the csv files for each algorithm are also saved in a separate *Results* folder.
  - For some of the algorithms there are also images created to show the tracing that has been performed. This can be used to visually review the images (see image 6).

Each algorithm comes with a set of adjustable parameters, and the results will vary depending on the selected method. When the tool is opened, each parameter has a default value, along with a brief description and the unit of measurement. Users can modify these parameters to fine-tune the results. It is important to always visually check the impact of any adjustments.

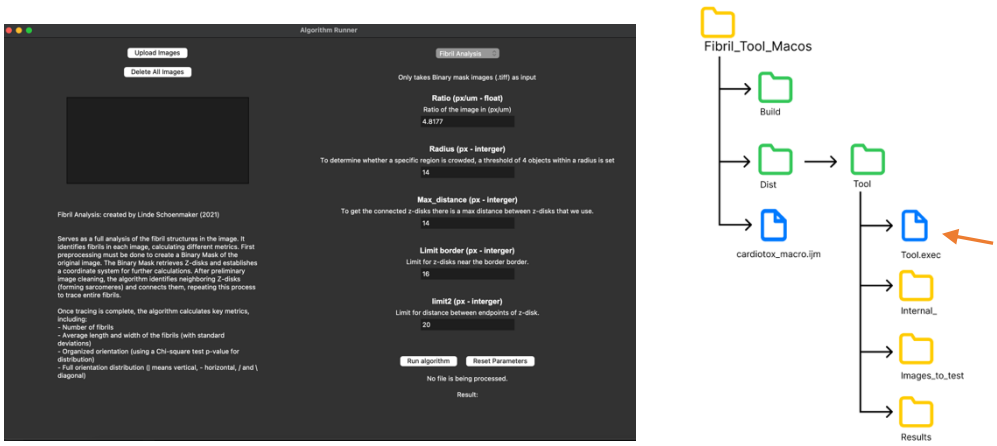


Image 5: Left, image of the program UI. Right, representation of the folder structure. The arrow shows the file which opens the program.

3) THE RESULTS

- After running the algorithm, a CSV file with the results will open automatically. Be sure to save and store this file, as running the algorithm again will overwrite it with new data.

- For the Fibril and Sarcomere tracing analyses, visual output files are also generated. As shown in Figure 6, the tool's file structure includes a “Results” folder, which contains separate subfolders for images generated from the Fibril and Sarcomere tracing analyses.

  In Figure 7, you can see examples of the output images. These images can serve as a reference when adjusting parameter settings, helping to guide your analysis and optimize results.

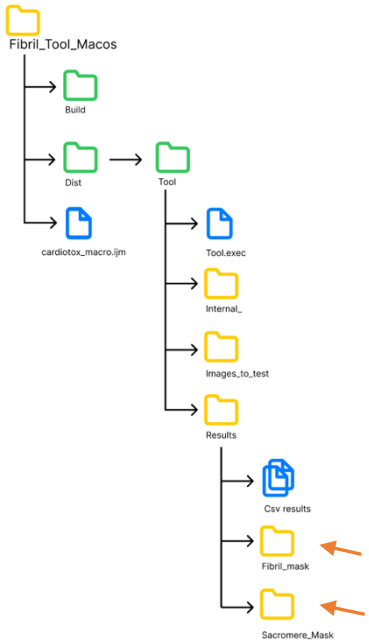


Image 6: Representation of the folder structure. The arrows show the location of where the generated images will be saved.

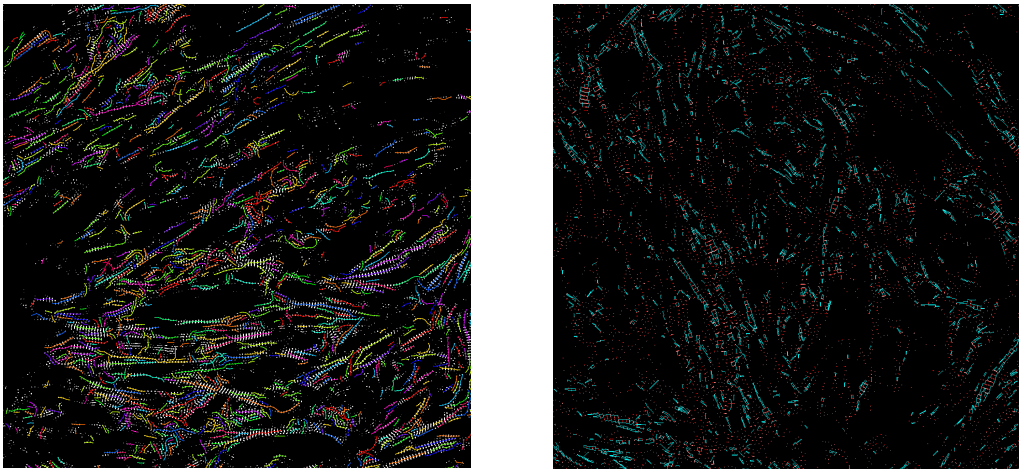


Image 7: Example images generated by the methods. The left image shows fibril tracing produced by the Fibril method, while the right image shows sarcomere tracing from the Sarcomere tracing method.

REFERENCES

[1] Efford N (2000). Digital Image Processing: A Practical Introduction Using Java. Boston: Addison-Wesley

[2] Lu Cao, Linde Schoenmaker, Simone A. Ten Den, Robert Passier, Verena Schwach, and Fons J. Verbeek. Automated Sarcomere Structure Analysis for Studying Cardiotoxicity in Human Pluripotent Stem Cell-Derived Cardiomyocytes. Microscopy and Microanalysis, 29(1):254–264, 12 2022.

[3] Tijmen van Wel. Novel way for the calculation of sarcomere length in human pluripotent stem cell derived cardiomyocytes.

Link to GitHub Page:
https://github.com/Mar2836890/Fibril_tool.git