Scratch Wound Assay Quantification Workflow

Author: Albert Wang

Last updated: 2025-07-19

Summary

This document outlines a MATLAB workflow for quantifying wound area and length from scratch wound assay images.

The scratch wound assay is a widely used in vitro method for studying cell migration. In this assay, a scratch or gap is created in a confluent monolayer of cells using a pipette tip or other tools. Researchers then observe how cells move into the gap over time. Under the microscope, the scratch typically appears as a clear, cell-free region between two dense regions of cells, which gradually closes as the cells migrate inward.

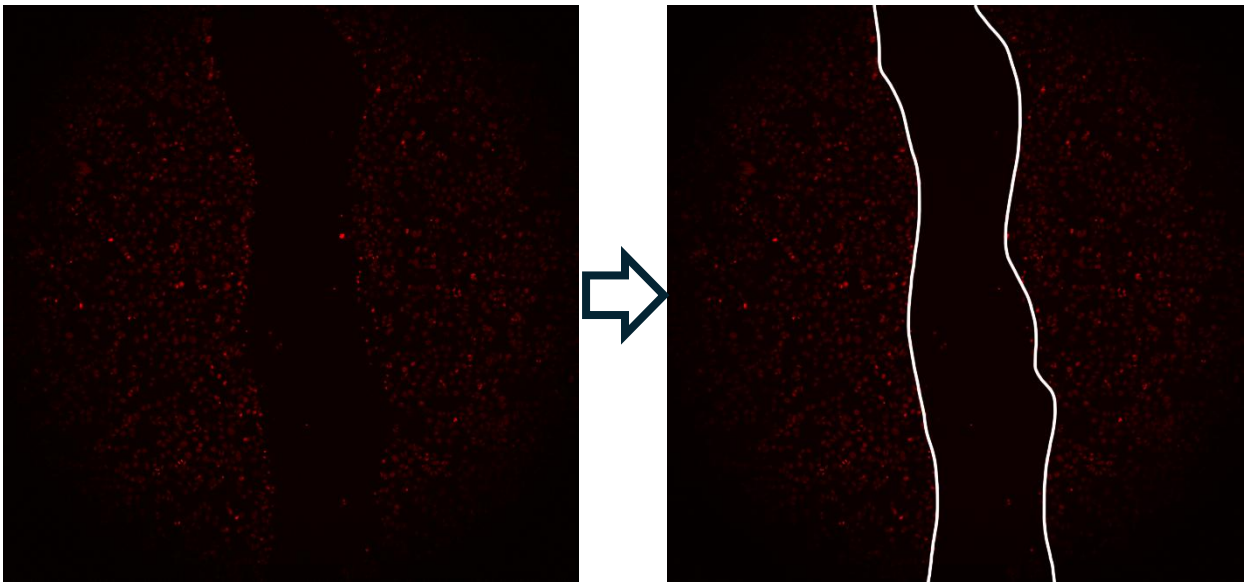
This workflow is designed for use with images where the wound region is bounded by two manually drawn white lines (see examples below). These boundaries can be in either vertical or horizontal orientation and are assumed to represent the physical edges of the cell-free (wound) area in a confluent monolayer of cells.

The script thresholds and skeletonizes the white boundaries, detects their endpoints, groups endpoints by boundary, and connects them to form a closed quadrilateral region of interest (ROI). It then fills the ROI and calculates the enclosed wound area in pixels. The script also computes the distance between the midpoints of the two connecting edges (those automatically drawn to close the ROI), providing a consistent measure of wound length. This length can be used for normalization or cross-sample comparison. Optional cropping can be applied to remove unwanted edge regions (rows or columns) from the image prior to processing.

Each image is processed individually, and the output includes an annotated image with the ROI highlighted and the wound length line overlaid in red. A summary Excel file is generated at the end, compiling the area and length values for all analyzed images.

Workflow Steps

Before running the script, users must manually draw two white lines (or black lines if using images with white background and adjust the thresholding condition accordingly) onto each image to mark the opposing edges of the scratch wound. For example:



1. Image Selection and Input

- Use uigetfile to select one or more RGB TIFF images.
- Prompt the user to specify scratch orientation (vertical or horizontal) and optional cropping amounts.
- If cropping is applied, the script will remove pixels from the top/bottom (vertical orientation) or left/right (horizontal orientation) sides of each image.

2. Preprocessing and Binary Line Extraction

- Convert each RGB image to grayscale using rgb2gray.
- Threshold the grayscale image to isolate white lines, generating a binary mask.
- Apply morphological thinning (bwmorph) to skeletonize the boundary lines.

3. Endpoint Detection and Boundary Assignment

- Use bwmorph('endpoints') to detect endpoints in the skeletonized lines.
- Identify and assign endpoints to two boundary groups using connected component labeling (bwconncomp, labelmatrix).
- Detect and match the appropriate endpoint pairs across the boundaries.

4. ROI Formation and Filling

- Connect paired endpoints from opposing boundaries using a custom drawline function.
- Combine the original skeleton with the connecting lines to form a closed polygon.
- Use imfill to fill the enclosed wound region.

5. Area Calculation and ROI Visualization

- Calculate the wound area as the number of filled pixels.
- Highlight the ROI in the original image using a blue overlay.

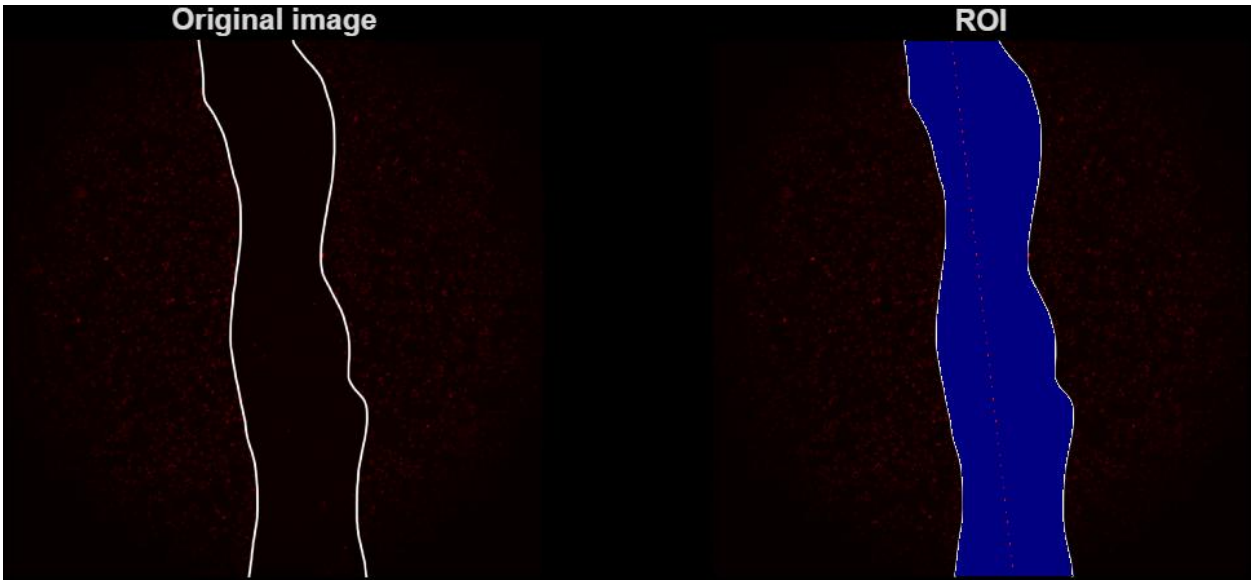
6. Wound Length Measurement

- Compute the midpoints of the endpoint connections.
- Measure the distance between these midpoints.
- This distance serves as the wound length, enabling normalization across images.
- Create a red line connecting the midpoints of the two connecting boundaries.

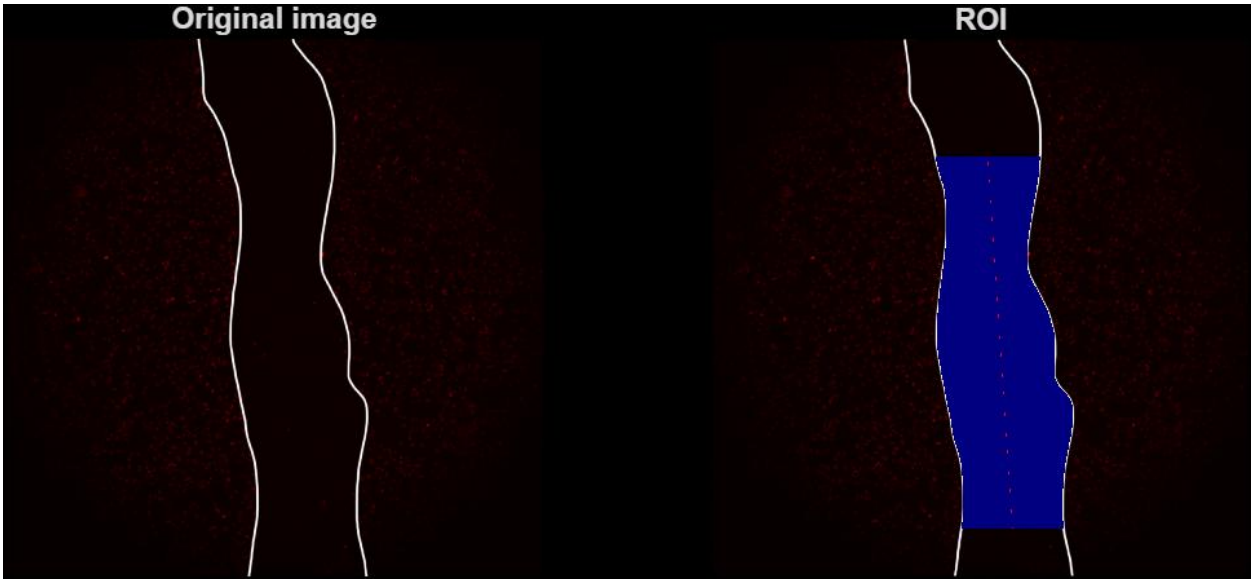
7. Result Display and Export

- Export each annotated image as a high-resolution TIFF with the ROI and length overlay.
- Compile the wound area and length for all images into a single Excel file using writetable.

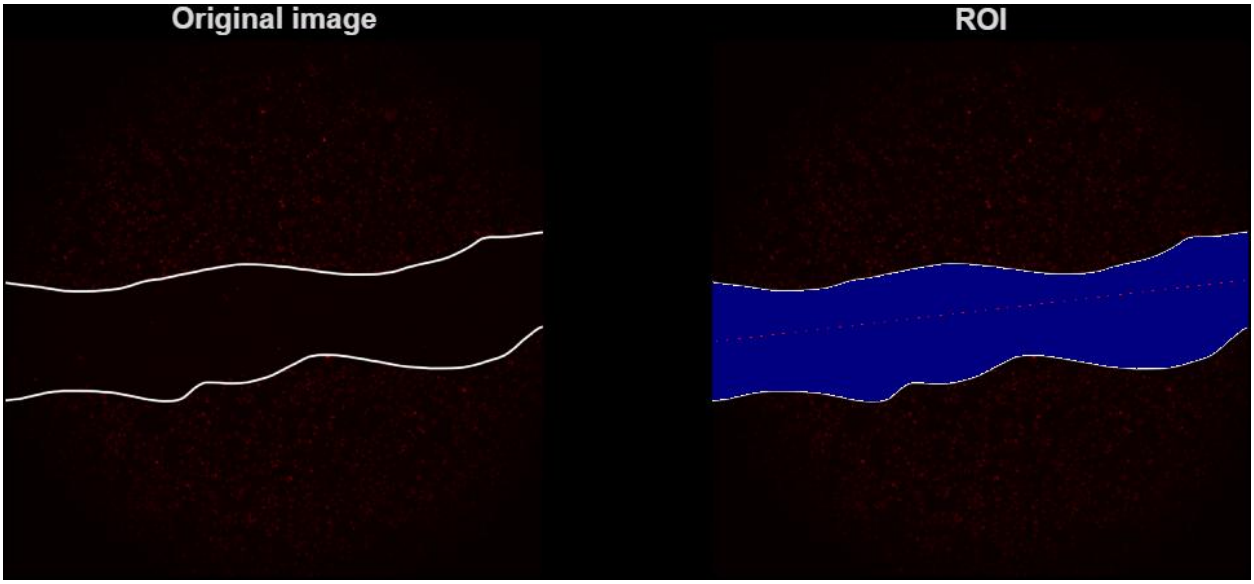
Example Outputs



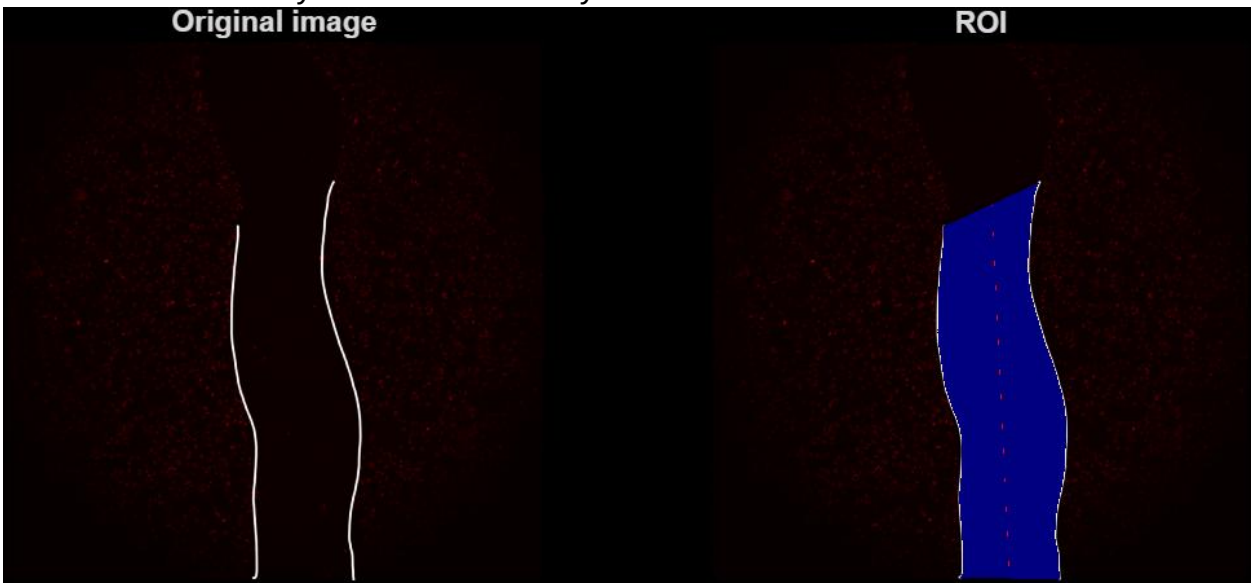
Crop 500 pixels from the top and 200 pixels from the bottom of the image.

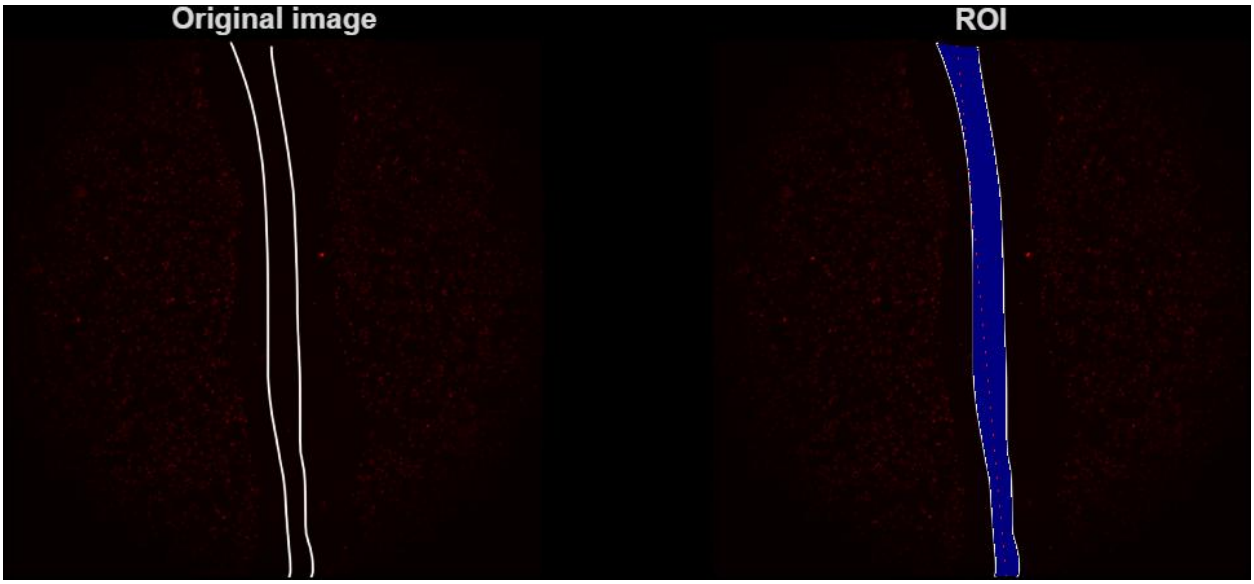


Horizontal scratch orientation



The ROI is defined by the user's manually drawn lines.





Notes

- This workflow assumes manually drawn white boundaries are present and distinct in each image.
- Only the first four endpoints are processed; images with extra lines or broken skeletons may produce inaccurate results.