# MATLAB AUC Analysis

% Add Bio-Formats toolbox to MATLAB path

addpath('C:\Program Files\MATLAB\R2024b\bfmatlab\bfmatlab'); % Update with your actual path

% Ensure Bio-Formats is correctly configured

bfCheckJavaPath();

bfCheckJavaMemory();

% List of ND2 file paths

nd2FilePaths = {

'C:\university o alabama data 20241203\DATA\transdermal prject\DATA\confocal\rat admin pen 20250215\site 2\1001.nd2',

'C:\university o alabama data 20241203\DATA\transdermal prject\DATA\confocal\rat admin pen 20250215\site 2\1002.nd2',

'C:\university o alabama data 20241203\DATA\transdermal prject\DATA\confocal\rat admin pen 20250215\site 2\1003.nd2',

'C:\university o alabama data 20241203\DATA\transdermal prject\DATA\confocal\rat admin pen 20250215\site 2\1004.nd2',

'C:\university o alabama data 20241203\DATA\transdermal prject\DATA\confocal\rat admin pen 20250215\site 2\1005.nd2',

}; % Add more file paths as needed

% Initialize variables for merged Z-stack and AUC tracking

stackAUCs = []; % To store AUC for each stack

stackSkinThickness = []; % Actual skin thickness per stack

cumulativeSkinThickness = []; % Cumulative skin thickness

currentTotalThickness = 0; % Initialize cumulative thickness

% Loop through each ND2 file

for fileIdx = 1:length(nd2FilePaths)

nd2FilePath = nd2FilePaths{fileIdx};

% Initialize the Bio-Formats reader

reader = bfGetReader(nd2FilePath);

% Get metadata

omeMeta = reader.getMetadataStore();

numZSlices = omeMeta.getPixelsSizeZ(0).getValue(); % Number of Z slices per stack

zStepSize = omeMeta.getPixelsPhysicalSizeZ(0); % Z-step size (µm)

% Convert Z-step size to numeric value

if ~isempty(zStepSize)

zStepSize = double(zStepSize.value());

else

zStepSize = NaN; % Handle missing metadata

end

% Compute skin thickness as: Thickness = Z-Step Size \* Number of Slices

skinThickness = numZSlices \* zStepSize;

% Get image dimensions

width = omeMeta.getPixelsSizeX(0).getValue();

height = omeMeta.getPixelsSizeY(0).getValue();

% Preallocate an array for the Z-stack

zStack = zeros(height, width, numZSlices, 'double');

% Loop through Z-planes and extract each one

for z = 1:numZSlices

zStack(:, :, z) = bfGetPlane(reader, z);

end

% Close the reader

reader.close();

% Normalize Z-stack for visualization (scale to [0, 1])

zStackNorm = (zStack - min(zStack(:))) / (max(zStack(:)) - min(zStack(:)));

% Compute mean intensity for each Z-plane

meanIntensities = mean(reshape(zStackNorm, [], numZSlices), 1);

% Compute AUC using trapezoidal integration

aucCurrentStack = trapz(1:numZSlices, meanIntensities);

% Store the AUC and actual skin thickness for this stack

stackAUCs = [stackAUCs; aucCurrentStack];

stackSkinThickness = [stackSkinThickness; skinThickness]; % Store actual thickness

% Compute cumulative skin thickness

currentTotalThickness = currentTotalThickness + skinThickness;

cumulativeSkinThickness = [cumulativeSkinThickness; currentTotalThickness]; % Store cumulative thickness

end

% Ensure unique x-values for the bar plot

[uniqueThickness, uniqueIdx] = unique(cumulativeSkinThickness, 'stable');

uniqueAUCs = stackAUCs(uniqueIdx);

% Plot the AUC for each stack

figure;

bar(uniqueThickness, uniqueAUCs, 'FaceColor', 'b', 'EdgeColor', 'k');

xlabel('Cumulative Skin Thickness (\mum)', 'FontSize', 14, 'FontWeight', 'bold');

ylabel('Area Under Curve (AUC)', 'FontSize', 14, 'FontWeight', 'bold');

title('AUC for Each Z-Stack (Cumulative Skin Thickness)', 'FontSize', 14, 'FontWeight', 'bold');

grid on;

% Print AUC values in the command window

disp('AUC for each Z-stack with cumulative skin thickness:');

for i = 1:length(uniqueThickness)

fprintf('Cumulative Skin Thickness %.2f µm: Total AUC = %.4f\n', uniqueThickness(i), uniqueAUCs(i));

end