## Intro to Image Processing with MATLAB

Prerequisites. No prior programming experience is required.

| Day 1 of 2. Learn MATLAB Fundamentals | |
|---|---|
| **Overview of the MATLAB Environment** | **Objective:** Learn basic vocabulary and skills for working in the MATLAB environment.<br><br>• Technical computing workflow: Using MATLAB as an informatics platform for research<br>• Orientation to the MATLAB Environment: Plots, Apps, Workspace, Variables, Scripts, Files, Functions, Documentation<br>• Exercise: import a table and explore, write output<br>• Exercise: import an image and explore, write output |
| **Understanding Image Data** | **Objective:** Learn how images are represented or stored in computer memory in order to build the intuition for how to operate on images in subsequent sections.<br><br>• Review of how images are generated from scientific instruments<br>• Using histogram and the ***Imtool*** app to understand image data at the pixel level<br>• Image color spaces, RGB to grayscale image conversion<br>• Data classes for Images (Logical, RGB, grayscale, double, u/int8, 16, 4D) and interconversion<br>• Exercise: Use the ***Color Thresholder*** app, generate code, call from a script, explore parameters |
| **Image Processing Methods.** | **Objective:** Learn the basic operations commonly used in image processing workflows and how to accomplish these in MATLAB using apps and functions.<br><br>• Overview of typical image processing workflows, capabilities of the MATLAB image processing apps, and introduction to the strategy of developing workflows using app-based approaches.<br>• Image Preprocessing<br>  ○ Histogram operations (histogram equalization, histogram matching)<br>  ○ Image filtering, noise removal<br>  ○ Correcting background illumination<br>• Image Segmentation. Exercise: Use the ***Image Segmentation*** app to segment an image, generate code, call the code from a script, explore parameters using the script<br>  ○ Segmentation using thresholding on pixel intensity values<br>  ○ Watershed segmentation<br>  ○ Active contours, graph cut<br>  ○ Texture segmentation<br>• Morphological operations: dilation, erosion, opening, closing, structuring elements<br>• Edge detection<br>• Logical operations and applying masks to images |

| | |
|---|---|
| **Basics of Data Analysis in MATLAB** | **Objective:** Image processing workflows typically include steps to extract tabular information from images which is then analyzed using statistical methods. In this section, we learn the basics of working with tabular data in MATLAB, and an overview of statistical methods in MATLAB.<br><br>• Exercise: Use the ***Image Region Analyzer App*** to analyze an image processed by the **Image Segmentation app**, generate code, call the function from a script to generate a table from an image<br>• Introduction to working with Tables, Matrices, Structures<br>• <u>Exercise</u>: Perform image segmentation using tables of properties of the image<br>• <u>Exercise</u>: Apply statistical methods to the table and generate a custom plot |
| **Workflow 1: 2D Image Segmentation** | **Objective:** In this afternoon workshop we will apply all the skills learned so far to develop our own custom 2D image processing workflow.<br><br>• Load data into the MATLAB workspace<br>• Exploring and understanding your data<br>• Determining whether and in what order to apply common image processing operations<br>• Using MATLAB apps to generate code, augmenting with MATLAB functions or community code<br>• Putting it all together in a script<br>• Automating the script for batch analysis<br>• Working with the resulting tabular data<br>  ○ Generating tables<br>  ○ Working with tables<br>  ○ Applying simple statistical analyses to tables<br>• Generating publishable plots or figures |
| **Using and sharing code with the internal/ external community.** | **Objective:** There is an open-source community of innovators who build and contribute example code on top of MATLAB. Before we break for the evening, we will familiarize ourselves with this community and learn how to utilize this resource in our day to day work.<br><br>• MATLAB documentation<br>• MATLAB answers<br>• MATLAB Add-on Explorer and the MathWorks File Exchange<br>• Stack Overflow, GITHUB, and other communities |

| Day 2 of 2. Project-Based Learning: Apply Fundamentals to Develop Custom Workflows | |
|---|---|
| **Objectives:**<br>• Consolidate your understanding by applying what we have learned so far to two new use cases<br>• Use both the point and click and computational aspects of MATLAB together to solve problems.<br>• Develop and obtain examples which you can take directly back to your own research and apply immediately to be more productive. | |
| **Workflow 2: 3D Analysis** | **Objective:** In the morning workshop, we will use apps and approaches from 2D image processing to tackle a 3D analysis problem. We will also explore a second approach using 3D image processing functions. This example will use a 3D image from a CT scan.<br>1. Working with 3D data as multiple slices of 2D data relying on apps for code generation<br>2. Working with 3D data using MATLAB's 3D image processing functions<br>After learning some of the newer MATLAB capabilities for 3D image processing, the instructor will help the class outline the two strategies. The class will then split themselves into two groups based on which approach they want to apply. Individually or as teams they will develop a workflow, sharing code as needed. Before lunch if the teams have an answer, they will share it. If not, the instructor will provide code examples of how each of these two methods work which the participants can use going forward. |
| **Workflow 3: Object tracking in 2D** | **Objective:** In this afternoon workshop we will import 2D time lapse data from microscopy images, design and build a custom workflow, automate it across multiple files, perform statistical analysis of resulting tables, generate publication quality plots and videos.<br>• Load data into the MATLAB workspace<br>• Understanding your data<br>• Image registration – what is it, when is it needed, and four standard options<br>• Determine whether and in what order to apply common image processing operations<br>• Using MATLAB apps to generate code, augmenting with functions or community code<br>• Putting this together into a script and automating this<br>• Building your tracking algorithm<br>  ○ Generating Tables<br>  ○ Working with Tables<br>  ○ Applying Statistics<br>• Generating publishable plots<br>• Generating videos from images for conferences |
| **Importing proprietary data formats into MATLAB via Fiji /Bioformats.** | **Objective:** An optional section on importing proprietary microscopy data formats<br>• What is Bioformats?<br>• Convert your data into a form MATLAB can readily use<br>• Loading and saving multipage tif files in MATLAB |