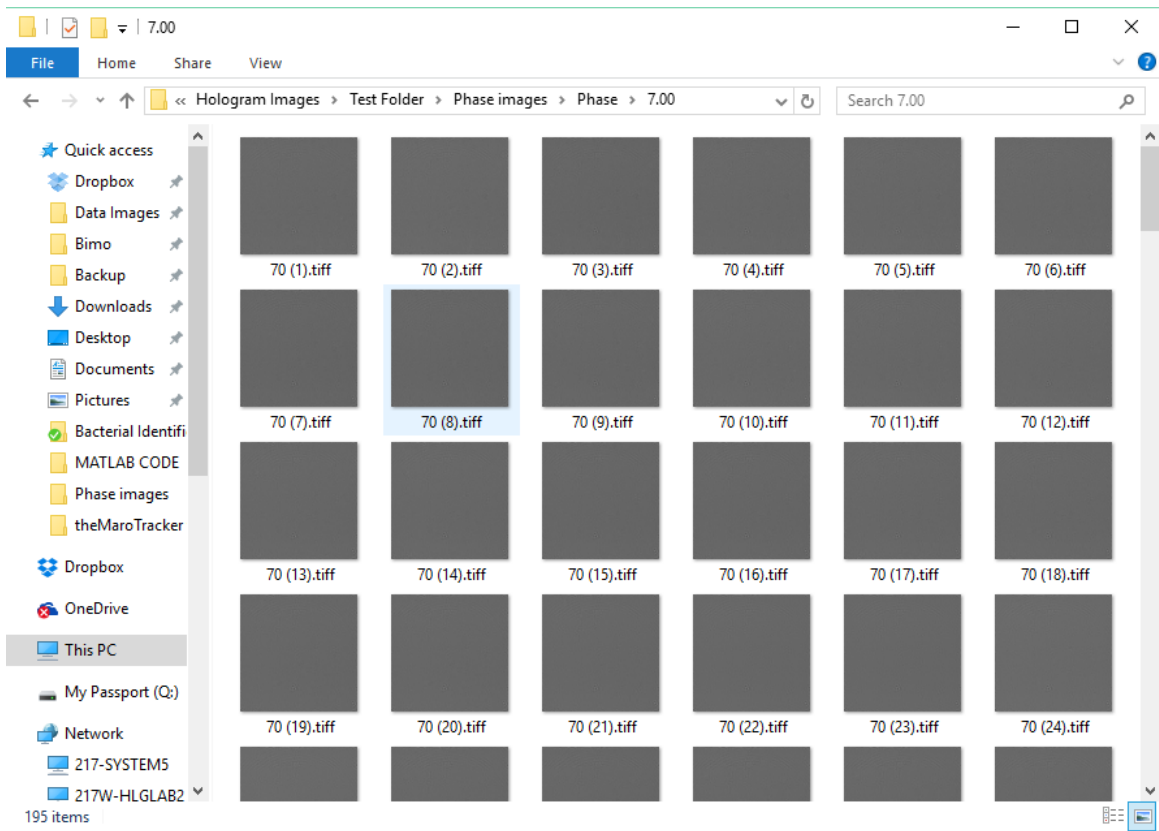# theMaroTrackerv1.x User Guide

1. Overview

1.1 Input:

Reconstructed Hologram Images.

Each folder contains a time series of images taken at a specific z-slice.

Each Image is labelled as 'int1 (int2).tiff', where int1 is an integer value representing the z-Slice number and int2 is in integer representing the time series number. An example is shown in *Figure 1* below.

Figure 1. Example of an input data folder

1.2 Output:

3D Tracks of the moving objects within the data set. An Example of the Program output is shown below in *Figure 2*.
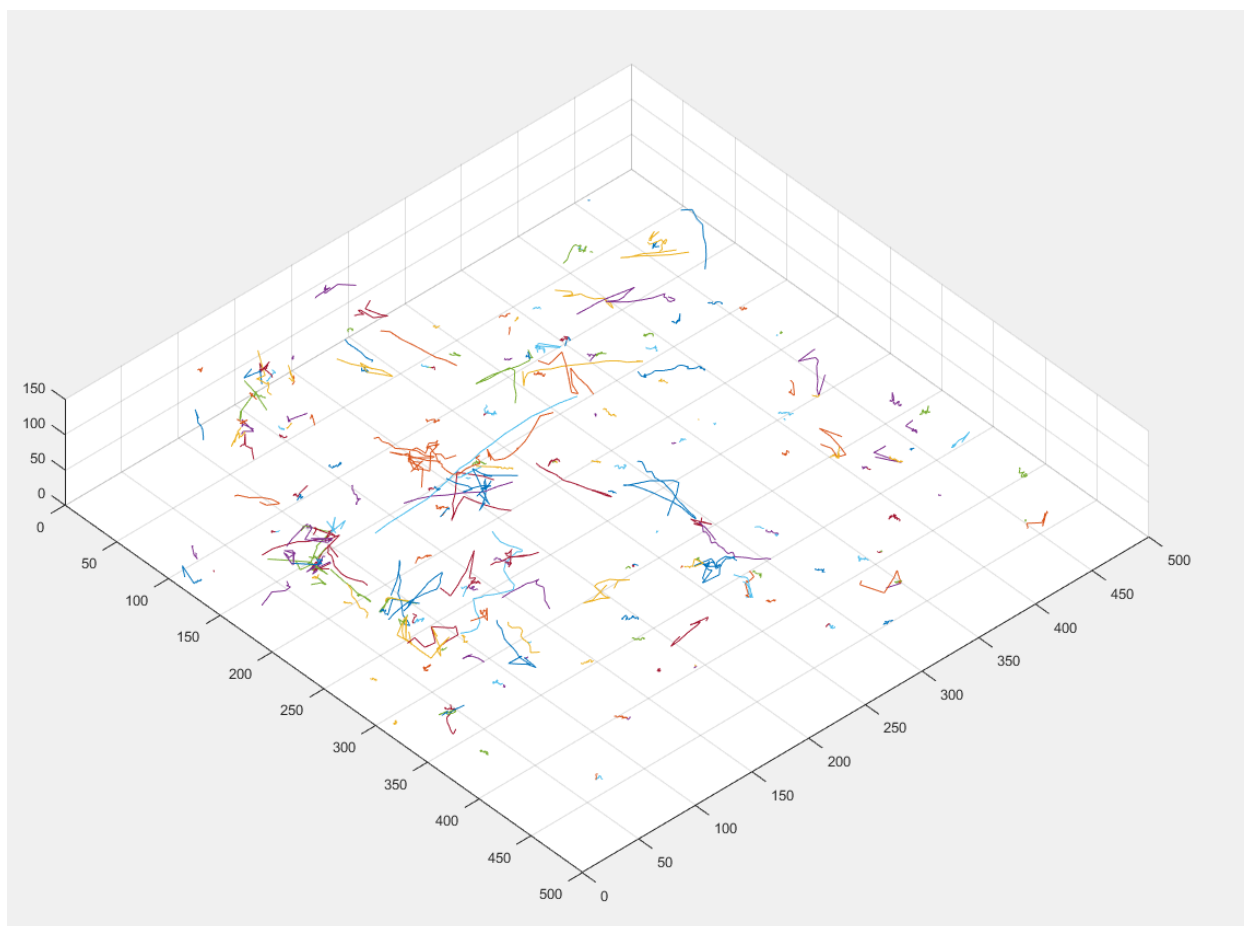
Figure 2. Tracking Output showing the paths taken by different bacteria in a sample.

1.3 Variables List:

Variable Name	Variable Location	Variable Description
zStart	preprocessingMain.m	Integer value to define the start of the data set. Int1 value explained earlier in this section
zEnd	preprocessingMain.m	Integer value to define the end of the data set. Int1 value explained earlier in this section
tStart	preprocessingMain.m trainingPhase1.m	Integer value to define the start of the data set. Int2 value explained earlier in this section
tEnd	preprocessingMain.m trainingPhase1.m	Integer value to define the end of the data set. Int2 value explained earlier in this section
numFrames	preprocessingMain.m	requests the user to enter the number of frames to be used in calculating the median image of the data set and each z-slice. This must be an odd number. Smaller numbers result in faster processing. Default value numFrames = 13.
imgScaleFactor	preprocessingMain.m	reduces the size of every image in the data set, and must be between 0 and 1. Default value imgScaleFactor = 0.25.

z	trainingPhase1.m	The specific zslice where training phase1.m will occur.
sections	trainingPhase1.m	is an integer which splits the image into sections^2 pieces to make identification more accurate. Default value is sections = 2, which would result in splitting the image into 4 quadrants
contrastFactor	trainingPhase1.m	enhances the contrast of the image after median subtraction to make the features more distinguishable from the background. Default value contrastFactor = 3
pCutoff	trainingPhase2.m trainingPhase3.m excecutionMainCode.m	determine the minimum value of y to be classified as bacteria
minCluster	trainingPhase2.m trainingPhase3.m excecutionMainCode.m	minimum number of connected pixels allowed in the matrix
max_linking_distance	trainingPhase4.m excecutionMainCode.m	determine the maximum velocity of a track by specifying the max distance that the centroid of a track is allowed to travel.
max_gap_closing	trainingPhase4.m excecutionMainCode.m	the maximum number of dropped frames in a track.

2. Running the Algorithm

2.1 Overview

This algorithm is broken up into 6 main methods; each method fulfilling a particular function.

The user must run each main method in the following order:

1. preprocessingMain.m
2. trainingPhase1.m
3. trainingPhase2.m
4. trainingPhase3.m
5. trainingPhase4.m
6. excecutionMainCode.m

At the start of every script, there are variables to be declared specific to each script (see section 1.3 – Variable List).

2.2 Method Descriptions

Method Name	Method Description
1. preprocessingMain.m	This code will find the median image of each z-slice in the data set. This median image will be

	subtracted from every other image within the same z-slice as a tool of noise reduction.
2. trainingPhase1.m	this code will train our classifier by presenting examples of bacteria at 1 z-slice for the user to select.
3. trainingPhase2.m	this code will help verify that the training has worked, and choose a larger data set to test the classifier
4. trainingPhase3.m	this code will help us select appropriate parameter value for pCutoff and minCluster.
5. trainingPhase4.m	this code will plot the tracks for us, once we are satisfied with our selection of parameters
6. excecutionMainCode.m	Final Execution - finding the tracks and plotting them in 3D.