



# KnotResolver: MATLAB Code Guide

---

KnotResolver is a MATLAB script that can be used to segment and track single filaments. The script is designed for a “beating assay” setup.

## Table of Contents

<b>Download source code</b>	<b>1</b>
<b>Prerequisites</b>	<b>1</b>
<b>1. Workflow</b>	<b>2</b>
1.1. Creating a ‘.csv’ file for analysis	2
1.2. Segmentation parameters	2
1.3. KnotResolver.m	3
1.4. Output	3
DemoOutput	3
AI*.pdf	3
MTdata.csv	4
resolveCoordinates_v*.mat	4
segOverlay_v*.tif	4
Tip angle2.pdf	4

## Download source code

The source code can be downloaded from GitHub <https://github.com/CyCelsLab/MTKnotResolver>

## Prerequisites

In order to use DICOT via source code, the user needs to install MATLAB with the following toolboxes:

- MATLAB 9.12
- Signal Processing Toolbox 9.0
- Image Processing Toolbox 11.5
- Statistics and Machine Learning Toolbox 12.3
- Curve Fitting Toolbox 3.7
- Parallel Computing Toolbox 7.6 (Optional)
- MATLAB Parallel Server 7.6 (Optional)
- Polyspace Bug Finder 3.6 (Optional)

The interface of KnotResolver requires **MATLAB version R2019b** or higher (Figure 1).

**Note:** Please close other figure tabs if open in MATLAB, before running DICOT.

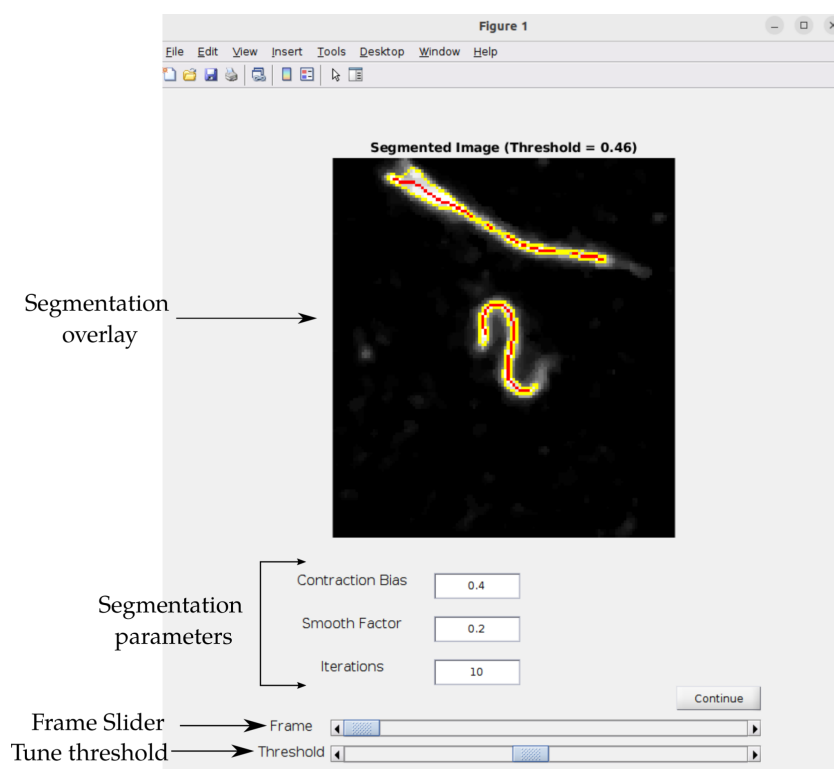


Figure 1. Graphical interface of the DICOT app. A simple interface to manually tune segmentation parameters. Inserted figure shows the input image with segmentation boundary (yellow) and filament segment (red) overlaid. Segmentation parameters can be tuned and the output can be checked for every frame using the frame slider.

## 1. Workflow

### 1.1. Creating a '.csv' file for analysis

The analysis begins by collating all time series containing a single microtubule pinned at one end. The details required in the '.csv' file headers are "FileName" and "FolderPath." These headers are case-sensitive and should reflect the name of the time series with extensions and the location of the parent folder. The code expects input image time series in 8-bit '.tif' format. An example '.csv' file and two default time series are available on GitHub [here](#).

Since different images can contain differences in noise and foreground, it is important to tune the parameters of segmentation. This can be done interactively using the "KnotSegmenter.m" file. Simply run the MATLAB file and tune the parameters until satisfied with the segmentation results.

The output is a '.csv' file with the suffix "optimized." This file contains the optimized segmentation parameters that can be run using the "KnotResolver.m" file. This '.csv' will be saved in the same directory as the original '.csv' file.

### 1.2. Segmentation parameters

**Threshold:** Threshold tuning between 0 and 1. Used as an input for the imbinarize intensity threshold method.

**Contraction Bias and Smooth Factor:** Input parameters for the activecontour method. Contraction bias ranges from -1 to +1. Positive values force the contours to shrink, and vice versa. The smooth factor determines the smoothness of the optimized curve.

**Iterations:** Number of iterations for the activecontour method.

### 1.3. KnotResolver.m

The ".csv" with the optimized parameters can be run directly with the KnotResolver.m script. For each file, the code shows the segmentation overlay on the original input image with a crosshair for MT selection. The central microtubule needs to be selected. The contour selection location determines the position of the fixed tip. Therefore, careful selection of the contour near the fixed tip should be made.

Once the selection of the central contour is made, the MT in the next frame is identified based on overlap with the initial selection, which is updated in every frame.

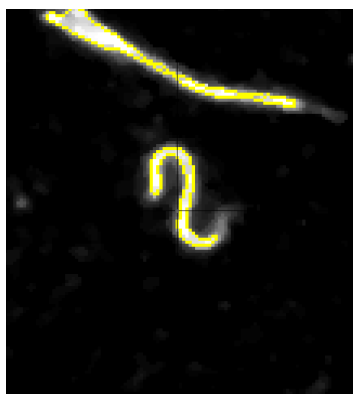


Figure 2. **KnotResolver figure overlay.** For each file in the “\*\_optimized.csv” the code prompts the user to select the contour. Here the selection should be made near the fixed tip which is assigned a fixed tip identity.

### 1.4. Output

#### *DemoOutput*

A “DemoOutput” is created if the InputFile\_optimized.csv file is used to run the KnotResolver.m script. The folder contains the segmentation output in a “.mat” file which has the same name as the input image. This “.mat” file contains the information of image size and the binary segmented skeleton. This file is created using the optimized segmentation parameters.

A separate folder with the same name as the input image is created and contains the following outputs:

#### *Al\*.pdf*

Contains a subplot of three figures arranged vertically. The first figure contains the overlay of the complete resolved time series color coded along the length. The second subfigure contains the kymograph of the tangent angles along the microtubule contours. The last subfigure contains the distance between the two ends of the filament in the time series.

***MTdata.csv***

Contains three columns that correspond to the frame number, X and Y coordinate of the skeletons. The indices are in pixels and need to be rescaled with the correct scaling factor.

***resolveCoordinates\_v\*.mat***

“.mat” file containing the resolved coordinates. This can be used for further analysis when required. The contour indices are arranged from tip to bottom.

***segOverlay\_v\*.tif***

Final output skeleton color coded along the length. This can be visually checked to confirm if the output is correct.

***Tip angle2.pdf***

FFT analysis of the tip angle suspended at the two tips of the beating microtubule.