Simulating Cortical Microtubule Dynamics in Realistic Trichome Shapes

Samuel Clucas [1], Dr Eashan Saikia [2], Dr François Nédélec [2]
1. Department of Biosciences, Durham University, UK
2. Sainsbury Laboratory, University of Cambridge, UK
With thanks to Robert Bellow (John Innes Centre, Norwich) for trichome microscopy images (figure 4).

Introduction and Aims:

My placement goal: to develop my computational skills in the field of plant biology.
I worked on "trichomesim" used to simulate cortical microtubule (MT) dynamics in complex cell geometries, building on Cytosim's simpler 2D/3D geometries. This involved:

- learning C/C++ to create an optimized nearest neighbour search (NNS) algorithm using a KD tree data structure to enhance computational performance.
- processing trichome confocal microscopy images to obtain quantitative density and orientation analyses to ensure model realism. This resulted in the creation of my own python program 'MT Plot'.

Visualising KD Trees & Nearest Neighbour Search (NNS) Logic:

The need for optimisation:

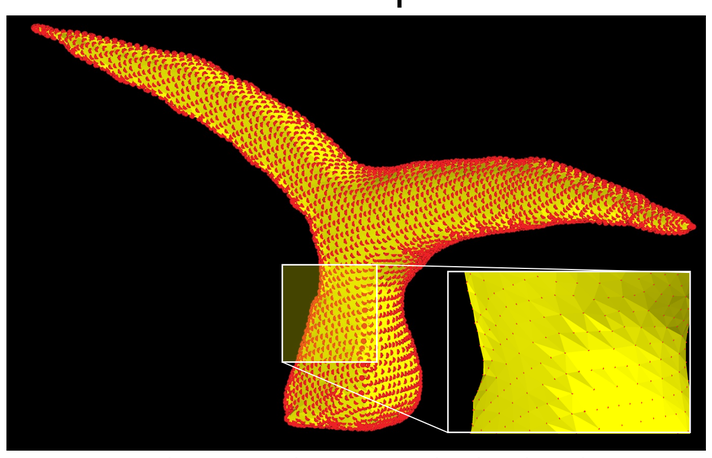

Figure 1. Trichome mesh as 3D node cloud, with almost 5000 nodes.

Frequent microtubule - mesh node positional comparisons necessitates faster alternative to sluggish brute-force search method.

2D Tree Construction:

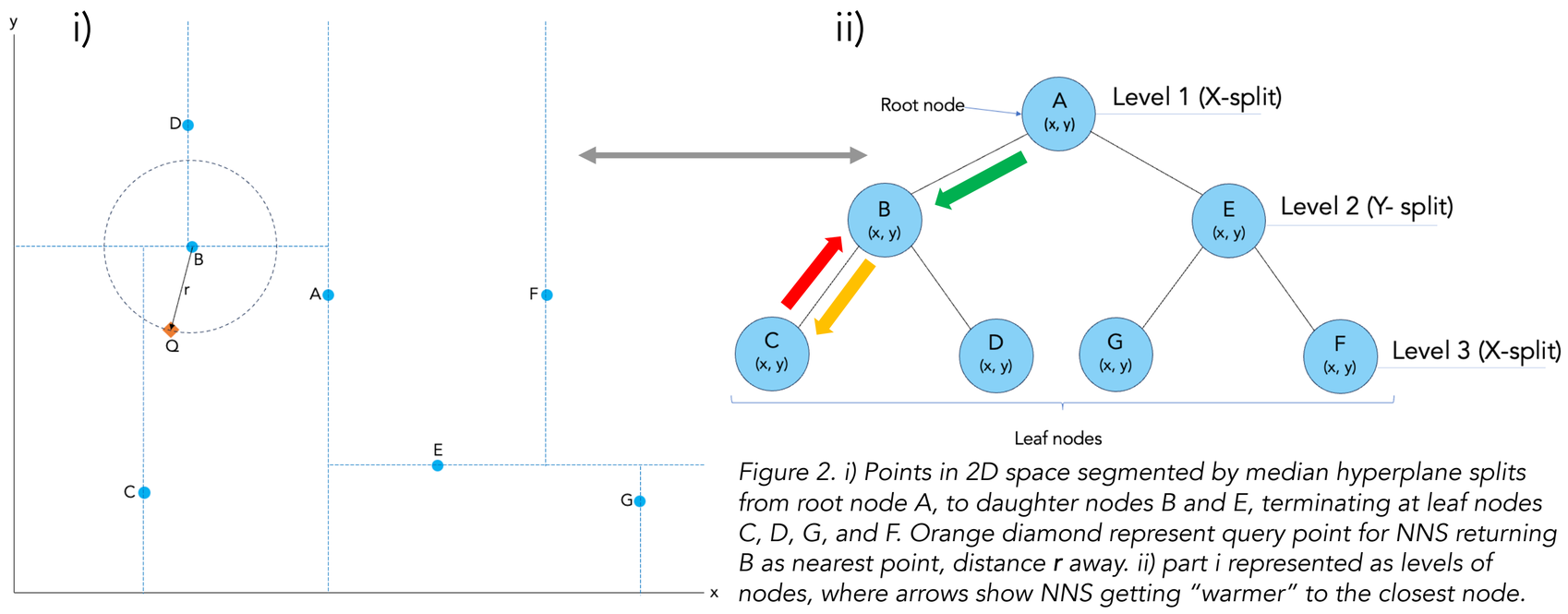

*Figure 2. i) Points in 2D space segmented by median hyperplane splits from root node A, to daughter nodes B and E, terminating at leaf nodes C, D, G, and F. Orange diamond represent query point for NNS returning B as nearest point, distance **r** away. ii) part i represented as levels of nodes, where arrows show NNS getting "warmer" to the closest node.*

Image Processing Workflows: MT Segmentation

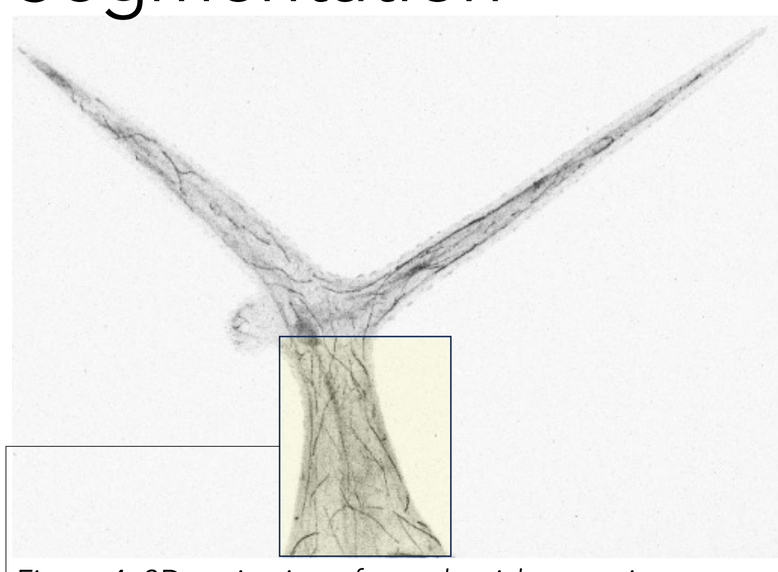

Figure 4. 3D projection of sample trichome microscopy images made in FIJI.

Figure 5. MT segmentation by ILASTIK pixel classification using machine learning

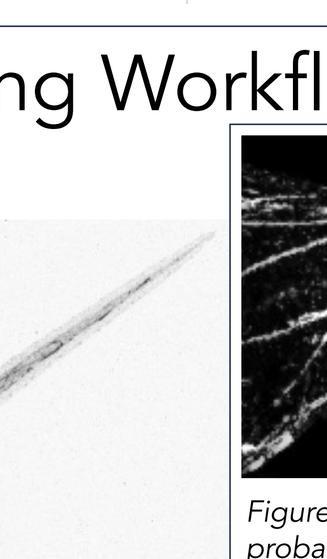

Figure 6. Z project of thresholded ILASTIK probability files in FIJI

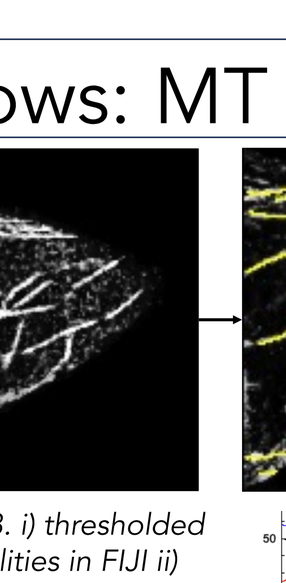

Figure 7. FIJI 3D Object Counter plugin.

Figure 8. i) thresholded probabilities in FIJI ii) automated spline fitting using "MOSAIC"[2] algorithms iii) plotted splines again using "MOSAIC"[2]

Search Time Prediction: 3D Tree vs. Linear

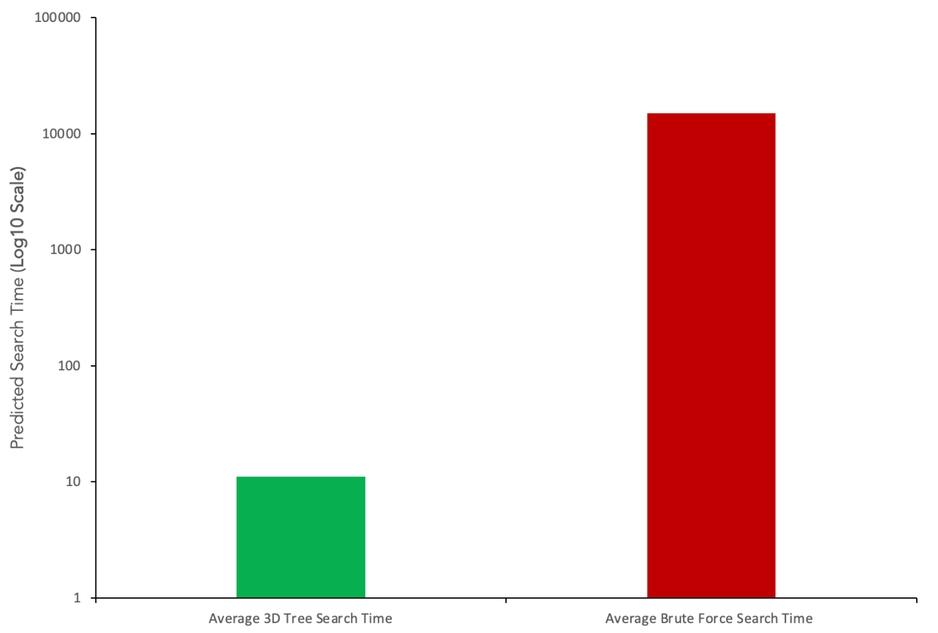

Figure 3. 3D tree NNS (green) vs. linear (red) average search time with $\log_{10}$ y-axis scale clearly demonstrating comparative efficiency.

Validating Automated Methods: Creating my own Program to Manually Label, Count, and Visualise Cortical Microtubules

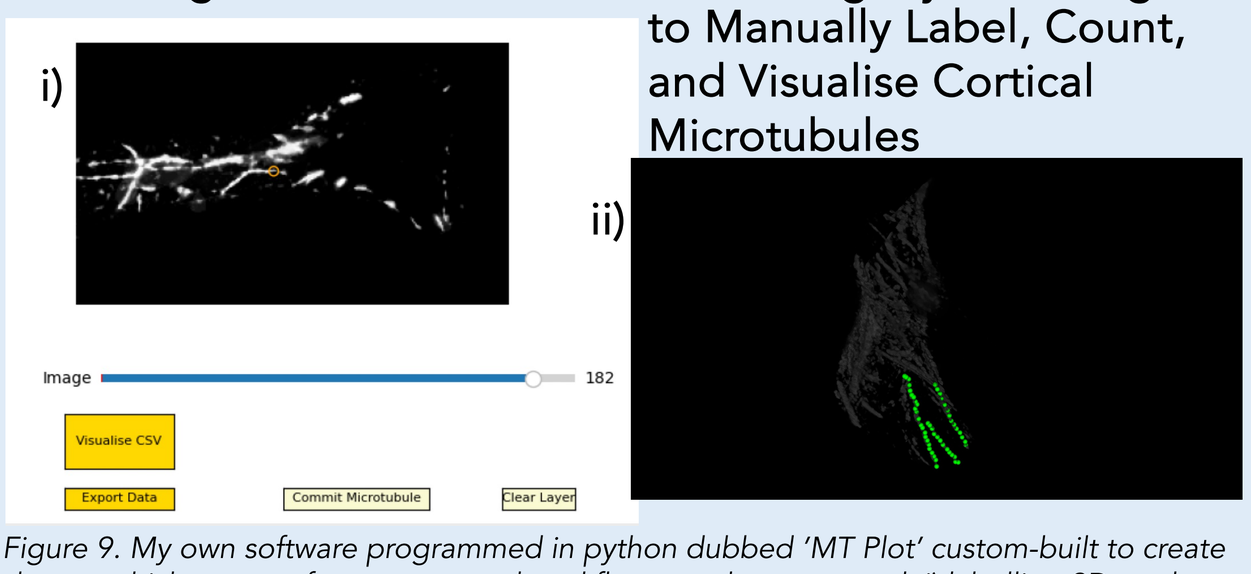

Figure 9. My own software programmed in python dubbed 'MT Plot' custom-built to create data to which outputs from automated workflows can be compared. i) labelling 2D stack 'slices' ii) 3D verification/visualisation.

References: 1. Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K. W., Schindelin, J., Cardona, A., & Sebastian Seung, H. (2017). Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. Bioinformatics, 33(15), 2424–2426.
2. Xiao X, Geyer VF, Bowne-Anderson H, Howard J, Sbalzarini IF. Automatic optimal filament segmentation with sub-pixel accuracy using generalized linear models and B-spline level-sets. Med Image Anal. 2016 Aug;32:157-72. doi: 10.1016/j.media.2016.03.007. Epub 2016 Apr 4. PMID: 27104582; PMCID: PMC5105836.