

Workshop structure

Part 1

Introduction

What are filopodia and how have they been studied

Part 2

Intro to Filopodyan workflow (Fiji)

A guided walk through the plugin for segmentation & tracking using a simple demo file

Part 3

Intro to Filopodyan workflow (R)

Phenotype comparison; correlations between properties; filopodium initiation; tip elongation

Part 4

a. Analysis: Phenotype comparison

Batch processing (Fiji) and downstream analysis

b. Analysis: Fluorescence & tip movement

Tip fitting, direction-corrected tip movement, cross-correlation analysis

Part 5:

Deconstruction

Deconstruction - some thoughts and ideas

Changes to Filopodyan code (Java):

Segmentation:

- tip assignment (edge case: bent filopodia)
- filopodium definition without ED (edge case: crossing filopodia)

Tracking:

- replace LinearAssigner with TrackMate

Changes to data analysis code (R)

Accessibility:

- ImageJ plugin --> script?
- use Filopodyan components from other languages & platforms?

Application

Modifying and Extending Filopodyan

FiloP

Implemented by Objects representing filopodia, [Filopodyan](#) uses [Filopart](#). Has many getter methods that could be implemented differently to store any process-like objects of interest.

Filopodyan uses [ArrayList<ArrayList<Filopod>>](#) throughout – a list of filopodia in a list of timepoints

FilopodyanPro

Implemented by Objects that process images to create binary masks, [Filopodyan](#) uses [LoGProcessor](#) and [ALTProcessor](#). These methods are flexible but could be replaced by any processing giving a signal mask for analysis. The [FilopodyanProcessor](#) is passed to [Filopodyan_.filopodia\(boolean preview, FilopodyanProcessor fp\)](#) to create the mask used for segmentation of filopodia.

LinearAssi

Implemented by Objects assigning filopodium identity over time, [Filopodyan](#) uses [OneStepAssigner](#). assign and cost methods could be implemented in any way required including calling [TrackMate](#) and setting track indexes from the [Model](#).

BoundaryAna

Interesting methods that Filopodyan only uses for visualisation. Not easily extendable, but contains code (under the GPL) for:

- local boundary signal and velocity measurement
- boundary signal and velocity kymographs
- cross correlation of boundary signal and velocity

Modifying and Extending Filopodyan

Filopodya

Manager and God Object. Handles flow control and calling of all other classes. Key methods that could be overridden to replace parts of the current pipeline are:

setImp(ImagePlus image)

important to call before anything else to assign the image to be analysed and get its dimensions

filopodia(boolean prev, FilopodyanProcessor filoProcessor)

called from FilopodyanGui, segments filopodia and growth cone bodies, applies LinearAssigner and passes FiloPods to a FiloFilter

call to run Filopodyan from your own class extending FilopodyanGui

override to do your own segmentation and tracking, and then use Filopodyan's filtering and track editor

filtered(ArrayList<ArrayList<FiloPod>> backPass)

takes the filtered tracks from FiloFilter, reruns LinearAssigner and creates a TrackEditor

call to pass in FiloPod tracks created by your own classes

override to take FiloPod tracks from Filopodyan and pass them to your own class

output(ArrayList<ArrayList<FiloPod>> backPass)

called from TrackEditor when the final FiloPod tracks have been created, measures filopodia and creates ResultsTables

call to display FiloPod tracks created by your own classes

override to change how results are output from Filopodyan