Briana Rocha-Gregg

UW Madison 2021

Abstract: The purpose of this matlab library is to perform in-vitro chemotaxis analysis on tracks generated with commercially available Imaris cell-tracking software. This package goes beyond the statistics generated by Imaris to quantify motility using a number of metrics including chemotaxis index, motility index, diffusion coefficient, motility class, distance from source inlet, pausing behavior etc. The functional outputs are organized, processed data structures that can be used for statistical analysis. Each structure includes measurements at the single-cell and population levels.

Pipeline diagram:

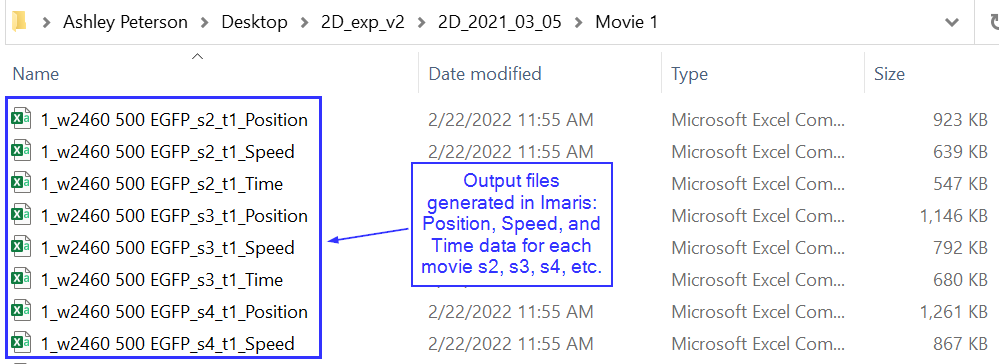
Before beginning, data should be organized in folders accordingly:

Experiment Title [ex: Arpc1bKO]

* Dataset title [ex. 2D]
  + Experimental Day in YYYY-MM-DD format
    - Movie or replicate title [ex. Movie 1]
      * Imaris spreadsheets for all positions within that replicate – the filename should include the replicate (s1, s2 etc.)

Example:

“C:\Users\anpet\Desktop\2D\_exp\_v2\2D\_2021\_03\_05\Movie 1”



Input:

There are 3 main components required to use the pipeline.

1. Imaris-generated csv files that contain at minimum: position and velocity information for each spot over time. Generally, Imaris allows two possible output formats. Either one spreadsheet with multiple sheets for each exported metric OR multiple .csv files (one file per metric) are generated. I have titled these two formats “compound” and “individual”, respectively. It is important to know the format of your .csv files and the corresponding title. Examples of each type are included in the example folder.
2. The user should generate a Data Key according to the provided template. During analysis, this key will be used to find data files and accurately label all outputs. In general, MATLAB doesn’t like spaces, -, or entries that begin with numbers. When making the key, keep this in mind to avoid potential errors during analysis. For experiments with multiple datasets (ex. 2D, 3D etc.), create a sheet per dataset. Note that a directory refers to the full-path where Imaris spreadsheets for all replicates within a given movie (slide) are located. If you imaged 6 positions of a slide, Imaris sheets for all 6 replicates should be in a single directory.

The columns should include:

Dir #: An integer – replicates from the same slide should all have the same directory and thus, directory #

Dirs: The full path where Imaris spreadsheets are located

Pos: Position within the slide – should not start with a number

Cell Line: Can be any label or identifier – avoid starting with a number and stay consistent. For example, don’t use CTL and Control on the same key. Otherwise, it will assume they are different cell lines

Stimulus: whatever chemoattractant or treatment is added to the cells – again, don’t start with a number and avoid spaces or dashes.

Text

Description automatically generated

1. Finally, the user should be prepared to input the following information and preferences into the template analysis script:
2. **masterPath**: full file-path to dataset directory entered as string (enclosed in single-quotes)
3. **src\_type**: ‘line’ or ‘point’ (point not fully tested for bugs) – refers to the chemoattractant source target. Use ‘line’ for Beebe lab slides
4. **process\_all**: 1 for yes; 0 for no. If you intend to process all replicates (most likely the case), enter 1. If you have not already screened all the movies and eliminated any with significant drift or blurriness, you can enter 0. However, I recommend doing this beforehand and removing any files you don’t intend to analyze. It’s less time and work this time because the code won’t attempt to read-in data you don’t intend to use. I usually put tossed replicates in a “junk” folder and store it somewhere else. These junk reps should also be excluded from the data key.
5. **timeInSec**: a number indicative of the frame-rate in seconds
6. **pixUnit**: a number that indicates the size of 1 pixel in uM
7. **numFrames**: number of frames per movie
8. **dataOrganization**: Enter 'compound' if you have one excel file with measurements on individual sheets, enter 'individual' if each measurement was exported to its own file
9. **min\_track\_length**: threshold for minimum tracklength for inclusion in analysis – I used 10

During the step for loading user inputs, the user will be asked to verify that wells are oriented correctly in reference to the original experiment. We found that newer versions of Imaris will arbitrarily invert some wells, which invalidates the tracks. Inverted wells must be specified at this step, or else the chemotaxis calculations will be incorrect.

The analysis pipeline is broken into 8 steps. For convenience, progression through each step is tracked and any errors that occur are logged for the user to review. Given the large number of replicates processed, the code was designed to continue despite errors that may occur. When errors occur, processing on that particular replicate is terminated, a report is generated, and analysis resumes on the next replicate. Replicates where errors occur are not included in the analysis output. However, users should carefully review error logs and verify the status of each replicate.

1. Day
2. Slide
3. Pos
4. Cell line
5. Stimulus
6. Id
7. Mean speed
8. Di
9. Cumulative distance
10. Ci
11. Accuracy
12. Mi
13. Cr
14. Num\_pauses
15. Pause-duration
16. vel\_free
17. vel\_speed

Matlab output deliverable folders:

1. Error Logs
2. Graphs
   1. This folder holds track images for all cells in the experiment
3. Variables
   1. <tab name>\_formatted4R (.csv file)
      1. This csv file contains the single cell track data for all replicates in the experiment along with values for each metric listed above. The address of this file will be the input for the R analysis in the next section.
   2. <tab name>\_expData\_<data> (Matlab structure)
      1. This structure contains important information for the experiment. It contains the following fields:
         1. masterPath: the file path entered in the first step of analysis
         2. statusTracker: the final status of each analysis step as outlined in the library header as well as how many errors were found during each step.
            1. note: if you completed the analysis in non-subsequent steps, some of the status columns may inaccurately show ‘Incomplete’
         3. ui: the remaining parameters entered by the user at the beginning of the analysis
         4. lut: the data key chosen at the beginning of analysis
         5. outputPaths: full file paths of the Variables, Graphs, and Error Logs folders
         6. processedDataLUT: look-up table separated out by each combination of drug/mutant/treatment
         7. R: the full output of single cell data for the experiment. This is the compiled Matlab data used to construct <tab name>\_formatted4R.csv
   3. <expData\_Final (Matlab structure)
      1. This is the final values of the \_expData
   4. processedData (Matlab file)
      1. This structure contains a field for each combination of conditions in the experiment. Then, all the data for that condition is separated into a structure for single cell and track data (singleCell) or for the track as a whole (cellMeans). In the singleCell structure, each line corresponds to In the cellMeans structure, each line corresponds to one cell.

R analysis

The included R analysis markdown performs three functions:

1. Generates plots for visualization of important cell motility metrics
2. Reports statistical significance based on a linear mixed effects model
3. Reports effect size of each motility metric grouped by cell line

We opted to use a linear mixed effects model because of the variability of the observed migration data.

Pre-requisites

Install the following packages in R prior to beginning:

* lme4
* lmerTest
* emmeans
* jtools
* purrr
* sjstats
* tufte
* sjplot

\*See supplement for detailed instructions to install

To use the analysis markdown, only one parameter is required to be changed. Navigate to the first chunk of code (labeled setup), and change **filepath** in the following line to the location of your csv file formatted for R. knitr::opts\_knit$set(root.dir = ‘**filepath'**)

Each chunk can be run separately for individual analyses for specific metrics, or the entire markdown can be run for a comprehensive analysis.

**Supplement**

**R package installation**

Type the following command into the R console, replacing <package\_name> with the package of interest: install.packages("<package\_name>"). Multiple packages can be installed at once by typing: install.packages(c("<package\_1>", "<package\_2"))

If using RStudio, packages can also be downloaded by navigating to the *Packages* tab and looking up each package. Click the Install button near the search bar