readDiaSessions.R Script Guide

Johns Hopkins University, Wu Lab

Author: Sun Jay Yoo

Date: July 5, 2017

# readDiaSessions

This script aims to take in a Diatrack .mat session file as input, along with several other user-configurable parameters, to return a track list of all the trajectories found in the session file. The naming scheme for each track is as follows:

[Last five characters of the file name].[Start frame #].[Length].[Track #]

(Note: The last five characters of the file name, excluding the extension, cannot contain “.”)

Input Parameters:

* *file* = file path to session file if known
* *interact* = option to interactively open window to select a file, by default set to TRUE
* *censorSingle* = option to remove and censor trajectories that do not have a recorded next/previous frame (trajectories that appear for one frame), by default set to TRUE
* *frameRecord*= option to add a fourth column to the track list after the xyz-coordinates for the frame that coordinate point was found (especially helpful when linking frames), by default set to TRUE
* *rowWise* = option to call to the *outputRowWise* function (see below), by default set to FALSE
* *colWise* = option to call to the *outputColWise* function (see below), by default set to FALSE
* *timer* = option to time the computation duration of the script

Sample Output:

|  |  |  |
| --- | --- | --- |
| > trackll <- readDiaSessions()  Reading Diatrack session file: HTZ1Halo\_fr10ms\_120mW\_5.mat ...  Session file read and processed.  > trackll[1:5]  $`0mW\_5.1.7.1`  x y z frame  1 26.90 84.34 1 1  2 26.75 84.15 1 2  3 25.97 84.31 1 3  4 26.64 84.10 1 4  5 27.31 84.04 1 5  6 25.93 84.37 1 6  7 26.06 84.33 1 7 | $`0mW\_5.1.4.2`  x y z frame  1 62.97 95.19 1 1  2 63.25 94.90 1 2  3 63.77 94.81 1 3  4 62.96 94.77 1 4  $`0mW\_5.1.13.3`  x y z frame  1 90.26 70.80 1 1  2 90.29 70.88 1 2  3 90.64 71.04 1 3  4 90.21 72.96 1 4  5 89.98 73.37 1 5  6 89.73 73.24 1 6  7 90.20 72.09 1 7  8 89.70 71.09 1 8  9 89.38 70.96 1 9  10 89.69 72.88 1 10  11 89.77 72.91 1 11 | 12 89.97 73.60 1 12  13 89.72 73.20 1 13  $`0mW\_5.3.2.4`  x y z frame  1 45.13 64.20 1 3  2 45.82 64.13 1 4  $`0mW\_5.6.10.5`  x y z frame  1 45.28 64.08 1 6  2 45.05 64.33 1 7  3 45.81 64.24 1 8  4 45.33 64.15 1 9  5 45.80 63.97 1 10  6 45.32 64.06 1 11  7 46.01 64.67 1 12  8 45.15 64.80 1 13  9 45.08 64.91 1 14  10 45.14 64.90 1 15 |

# removeFrameRecord

This helper script aims to make track lists with a fourth frame record column backwards compatible with other smt functions that rely on track lists with only three columns for the xyz-coordinates. The fourth column is simply removed from the given track list.

Input Parameters:

* *track.list* = a track list output from readDiaSessions.R

Sample Output:

|  |
| --- |
| > trackll.removed <- removeFrameRecord(trackll)  > trackll.removed[1]  $`0mW\_5.1.7.1`  x y z  1 26.90 84.34 1  2 26.75 84.15 1  3 25.97 84.31 1  4 26.64 84.10 1  5 27.31 84.04 1  6 25.93 84.37 1  7 26.06 84.33 1 |

# getStartFrame

This helper script simply returns the starting frame of a track/trajectory (using its name) at a given index for a track list.

Input Parameters:

* *track.list* = a named track list output from readDiaSessions or readDiatrack
* *index* = index of the track in the track list (track number)

Sample Output:

|  |
| --- |
| > getStartFrame(trackll, 5)  [1] 6 |

# outputColWise

This script takes in a track list from readDiaSessions or readDiatrack (with or without the fourth frame record column) and transposes it into a .csv column-wise output, similar to a Diatrack output, in the current directory.

Input Parameters:

* *track.list* = a track list output from readDiaSessions or readDiatrack

# outputRowWise

This script takes in a track list from readDiaSessions (with the fourth frame record column) and creates a .csv row-wise output, similar to an ImageJ output, in the current directory.

Input Parameters:

* *track.list* = a track list output from readDiaSessions with *frameRecord* set as TRUE

# linkSkippedFrames

This script aims to link trajectories that seem to have skipped (or do not appear for) a number of frames. Given user input for a tolerance level to limit how far the next point after the skip can deviate from the last point in pixel distance and a maximum number of frame skips possible, all trajectories falling within these parameters are automatically linked, renamed, and ordered accordingly. The naming scheme for each linked track is as follows:

[Last five characters of the file name].[Start frame #].[Length].[Track #].[# of links]

(Note: The last five characters of the file name, excluding the extension, cannot contain “.”)

Input Parameters:

* *track.list* = a named track list output from readDiaSessions or readDiatrack (Note: although not required, in order for the output to have a frame record column, the input must have one as well)
* *tolerance* = distance tolerance level measured in pixels after the frame skip
* *maxSkip* = maximum number of frames a trajectory can skip (eg. if the *maxSkip* for a trajectory ending in frame 7 was 3, the next linked trajectory can start up to a maximum frame of 11)

Sample Ouput:

|  |  |  |
| --- | --- | --- |
| > trackll.linked <- linkSkippedFrames(trackll, tolerance = 5, maxSkip = 10)  Linking trajectories with a tolerance of 5 and a maximum frame skip of 10 ...  1123 links found.  > trackll.linked[1:3]  $`0mW\_5.1.7.1.0`  x y z frame  1 26.90 84.34 1 1  2 26.75 84.15 1 2  3 25.97 84.31 1 3  4 26.64 84.10 1 4  5 27.31 84.04 1 5  6 25.93 84.37 1 6  7 26.06 84.33 1 7 | $`0mW\_5.1.20.2.4`  x y z frame  1 62.97 95.19 1 1  2 63.25 94.90 1 2  3 63.77 94.81 1 3  4 62.96 94.77 1 4  5 63.87 94.37 1 12  6 64.95 94.15 1 13  7 65.05 93.65 1 17  8 64.80 93.93 1 18  9 63.80 94.90 1 19  10 64.15 94.36 1 20  11 64.17 94.10 1 21  12 64.04 94.65 1 23  13 62.80 95.71 1 24  14 63.90 94.98 1 25  15 63.58 94.96 1 26  16 63.80 94.74 1 27  17 63.80 94.75 1 28  18 65.04 94.20 1 30  19 64.77 94.14 1 31  20 64.81 94.02 1 32 | $`0mW\_5.1.13.3.0`  x y z frame  1 90.26 70.80 1 1  2 90.29 70.88 1 2  3 90.64 71.04 1 3  4 90.21 72.96 1 4  5 89.98 73.37 1 5  6 89.73 73.24 1 6  7 90.20 72.09 1 7  8 89.70 71.09 1 8  9 89.38 70.96 1 9  10 89.69 72.88 1 10  11 89.77 72.91 1 11  12 89.97 73.60 1 12  13 89.72 73.20 1 13 |

# Testing (readDiaSessions)

readDiaSessions takes Diatrack .mat files as input, and returns a list of data frames (a track list) of all the particle trajectories. The aim is to optimize and un-censor this process, instead of having to use MATLAB to extract a large .txt file which is then fed into readDiatrack in R. Unlike the previous MATLAB script, this script has an option to un-censor, but is censored by default. Either way, this script resulted in slightly faster computation time (depending on the system) in testing.

A .mat session file with 10117 frames was used to test both scripts.

Using the previous MATLAB script, a 272.6MB .txt file was first created and was then fed into the readDiatrack() script to output track lists. Automating this process using "matlabr" resulted in 4488 censored tracks (should be 4487 tracks since the script does not censor first frame) in 3:48 mins.

Using readDiaSessions, the intermediate .txt file was no longer needed to be created and the session file directly results in track lists. This script resulted in 4487 censored tracks in 2:00 mins and 34689 uncensored tracks in 2:01 mins.