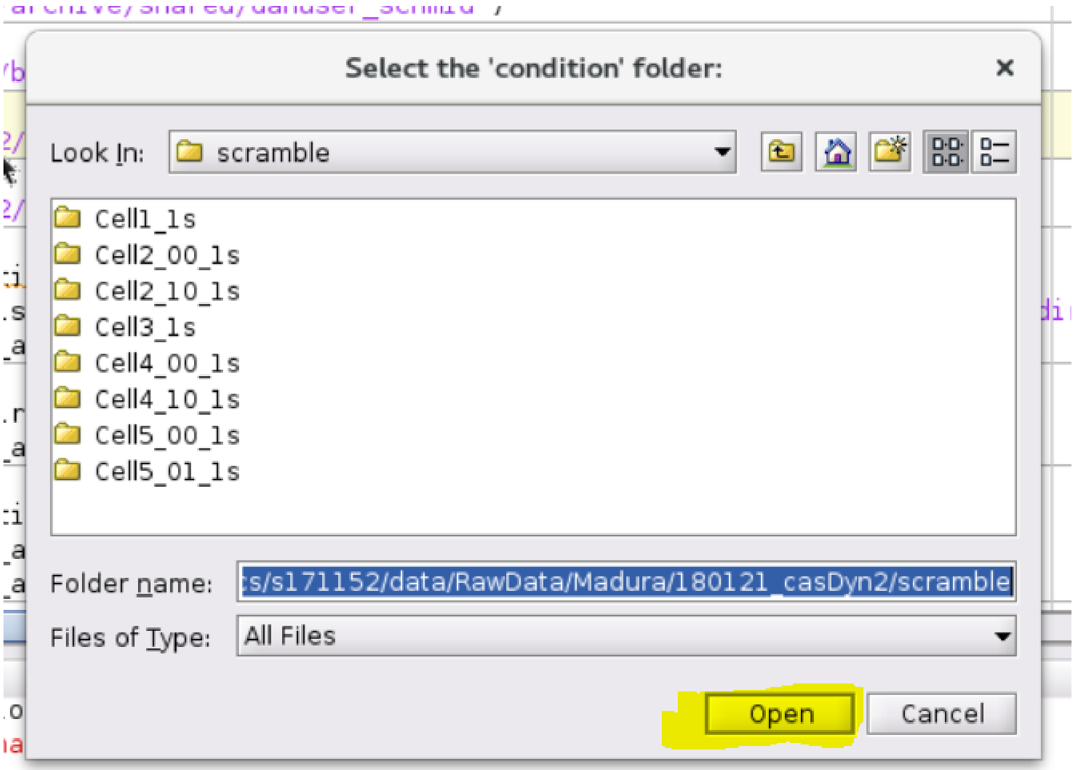
This software, “disassembly asymmetry score classification” (DASC), is based on the new method developed in the article, *“DASC, a sensitive classifier for measuring discrete early stages in clathrin-mediated endocytosis”*, by Wang X. *et. al. 2020*. DASC is built upon and embedded in cmeAnalysis, see [https://github.com/DanuserLab/cmeAnalysis.](https://github.com/DanuserLab/cmeAnalysis)

Quick start

1. This part is for intuitively using DASC to analyze single channel movies acquired on the same day. Before running DASC, to finish cmeAnalysis for these movies first is required.
2. First, ‘add to path’ the downloaded DASC package folder in Matlab. Then setup default parameters by the command line:

   ```
   pm = dasParameter();
   ```

3. Next, a window will pop out for selecting the folder where the control condition is contained. Click ‘open’ after finding the folder.

   Select the 'condition' folder:

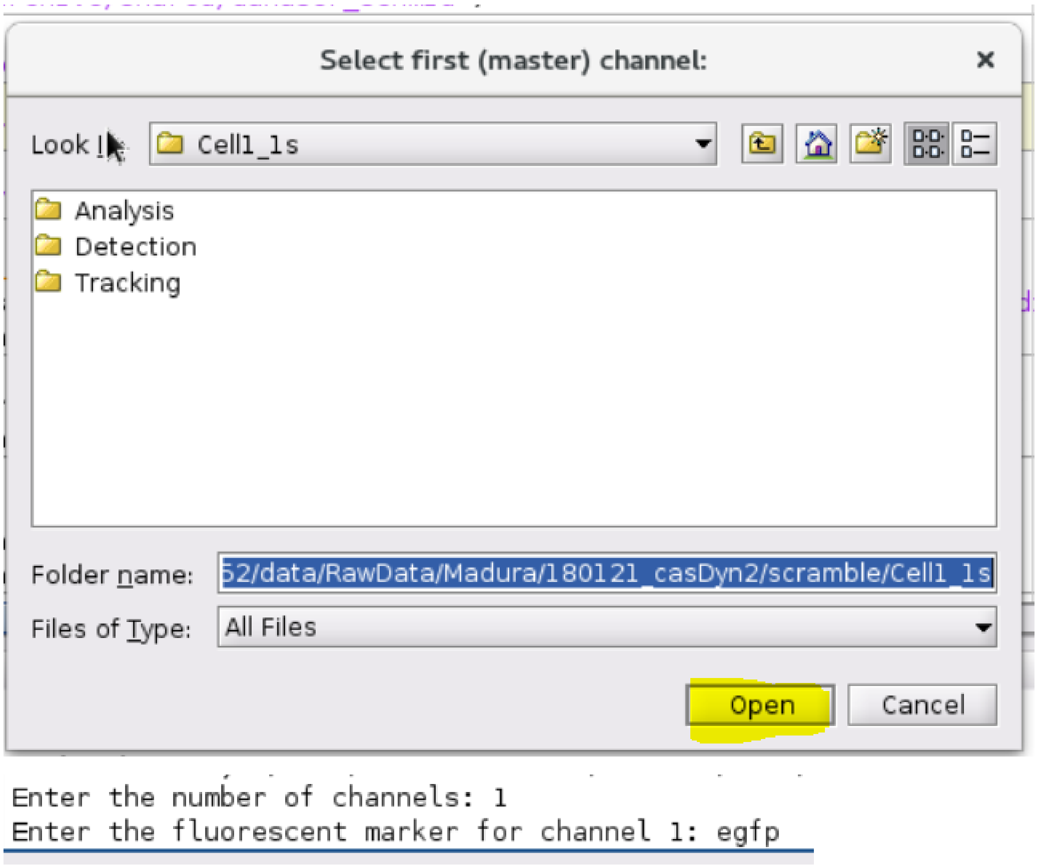
   Look In: scramble

   Cell1_1s
   Cell2_00_1s
   Cell2_10_1s
   Cell3_1s
   Cell4_00_1s
   Cell4_10_1s
   Cell5_00_1s
   Cell5_01_1s

   Folder name: s/s171152/data/RawData/Madura/180121_casDyn2/scramble

   Files of Type: All Files

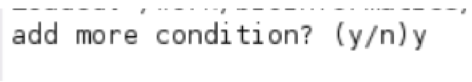
4. Next, “Enter the number of channels:” will show in the Matlab command window. Input accordingly. For example, enter 2 if control and one siRNA condition is considered. Then a folder containing ‘Detection’, ‘Tracking’ and ‘Analysis’ as the result of cmeAnalysis will pop out automatically. Click ‘open’ again. Also, input the fluorescent marker of the movies.



Folder name: 52/data/RawData/Madura/180121_casDyn2/scramble/Cell1_1s

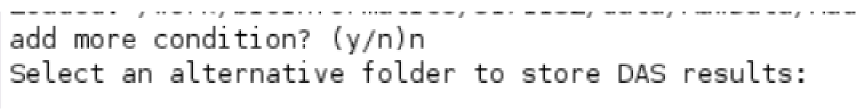
```
Enter the number of channels: 1
Enter the fluorescent marker for channel 1: egfp
```

5. Next, a question of whether to add more condition is shown, choose ‘y’ for more condition(s) or ‘n’ to terminate.



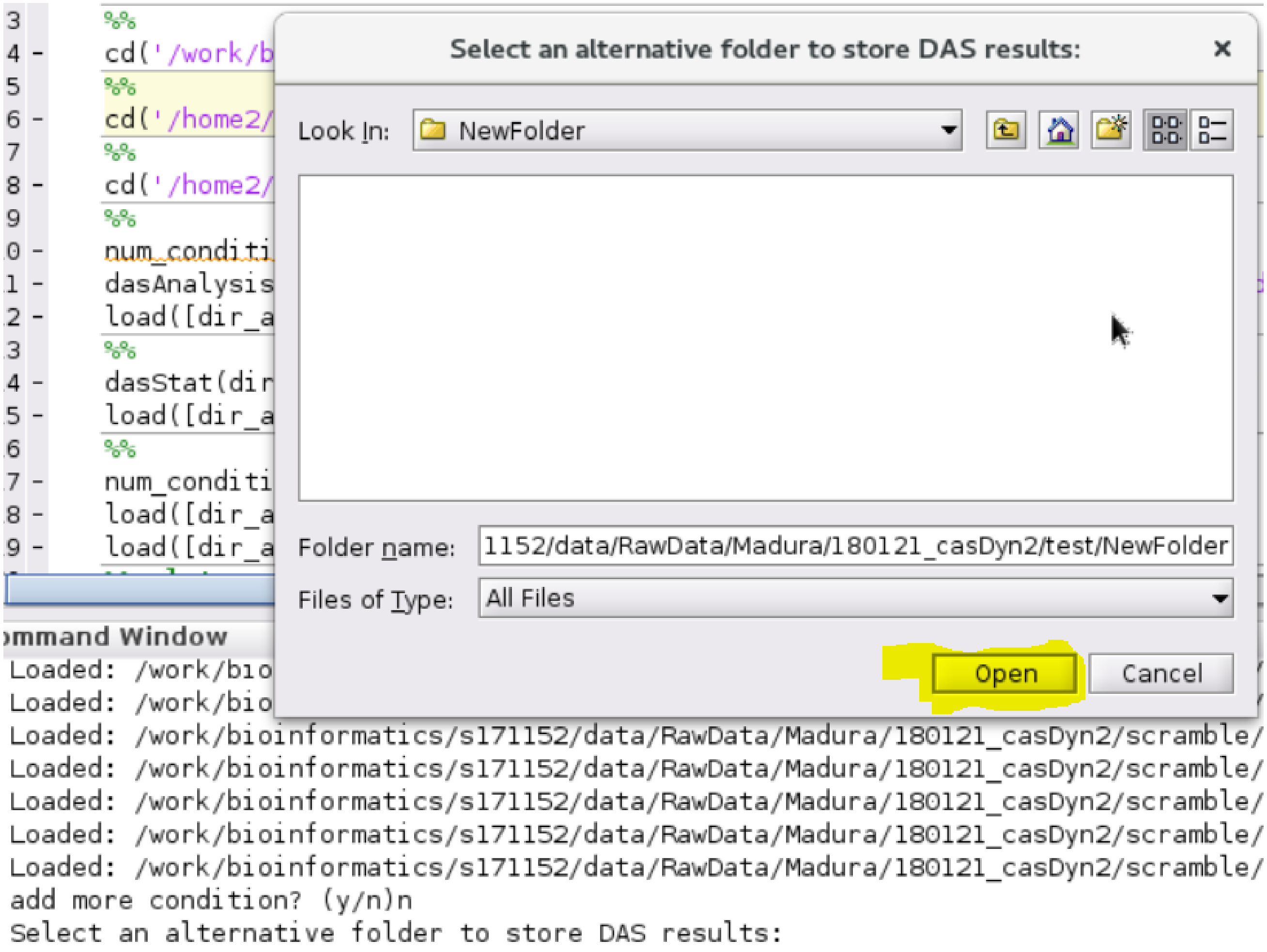
```
add more condition? (y/n)y
```

6. After all the conditions are selected (‘n’ in step 5), a request for choosing the result folder for DASC will be shown.



```
add more condition? (y/n)n
Select an alternative folder to store DAS results:
```

7. After selecting where the DASC result of the conditions will be located, click ‘open’.



```
%%
cd('/work/b
%%
cd('/home2/
%%
cd('/home2/
%%
num_conditi
dasAnalysis
load([dir_a
%%
dasStat(dir
load([dir_a
%%
num_conditi
load([dir_a
load([dir_a
```

Select an alternative folder to store DAS results:

Look In: NewFolder

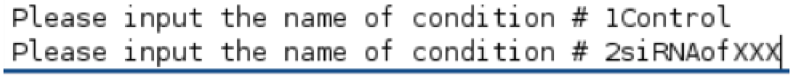
Folder name: 1152/data/RawData/Madura/180121_casDyn2/test/NewFolder

Files of Type: All Files

ommand Window

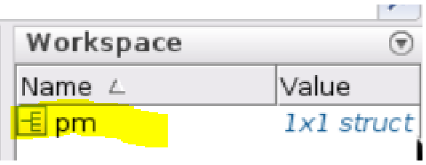
```
Loaded: /work/bio
Loaded: /work/bio
Loaded: /work/bioinformatics/s171152/data/RawData/Madura/180121_casDyn2/scramble/
Loaded: /work/bioinformatics/s171152/data/RawData/Madura/180121_casDyn2/scramble/
Loaded: /work/bioinformatics/s171152/data/RawData/Madura/180121_casDyn2/scramble/
Loaded: /work/bioinformatics/s171152/data/RawData/Madura/180121_casDyn2/scramble/
Loaded: /work/bioinformatics/s171152/data/RawData/Madura/180121_casDyn2/scramble/
add more condition? (y/n)n
Select an alternative folder to store DAS results:
```

8. Next, input the name of each condition.



```
Please input the name of condition # 1Control
Please input the name of condition # 2siRNAofXXX
```

9. Finally, the default parameter setting is finished. Now, there should be a variable ‘pm’ in the workspace.



Workspace

Name	Value
pm	1x1 struct

10. Last, use this command to run DASC with the parameter structure ‘pm’:

dasSingleCondition(pm);

Parameter setting

1. First, the above process can be simplified by manually setting up paths for movies and result to avoid pop-out windows:

   ```
   pm = dasParameter('data_all',data_all,'dir_alt',dir_alt,'con_name',con_name);
   ```

   where data_all is a cell variable indicating the directory of where the movies are stored, the frame rate of the movies, the length of the movies and so forth. To get this variable, use:

   ```
   data_all{1} = loadConditionData('/home/Folder/contains/Control/Movies', {"}, {'egfp'});
   data_all{2} = loadConditionData('/home/Folder/contains/siRNA_XXX/Movies', {"}, {'egfp'});
   ```

   If there are more conditions, do data_all{3}, data_all{4} and so forth. The function ‘loadConditionData’ is also described in cmeAnalysis instruction.

   Also, the path for DASC result is:

   ```
   dir_alt='/home/AnotherFolder/ForStoring/DASC_result';
   ```

   and a cell variable indicates the condition names:

   ```
   con_name={'Control', 'siRNA_XXX'};
   ```

   All of the 3 variables above need to be defined before getting ‘pm’. Note here that ‘con_name’ must have the same element number as ‘data_all’ has.

2. In addition, some other parameters can be changed for convenience. For example, using

   a.
   ```
   pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                   'save_image',false);
   ```

   Stops saving figures. Otherwise, figures are saved in dir_alt.

   b.
   ```
   pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                   'pdf_conf',false);
   ```

   Stops adding statistical confidence intervals (CI), which is slow.

   c.
   ```
   pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                   'plot_fig1',false);
   ```

   Stops plotting basic figures for control movies, which is unnecessary for phenotype comparison.

   d.
   ```
   pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                   'fig_dir', ' /home/path/for/figures');
   ```

   Allows storing figures in an alternative path defined by 'fig_dir'.

   e.
   ```
   pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                  'fig1_mosaic', true);
   ```

Adds frame by frame mosaic extracted from raw movies, which is a good quality measurement.

f.
```
pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                 'fig_disp_mod','none');
```
Stops plotting any figure.

3. The above setting can also be done in combination as, for example,
```
pm = dasParameter('data_all', data_all,'dir_alt',dir_alt,'con_name',con_name,...
                 ' plot_fig1',false, 'fig_disp_mod','none');
```

Dual channel movies

1. Setup parameters as in **Quick start** but choose channel number as 2, and select folders specified for each channel. See cmeAnalysis instruction for details. After getting ‘pm’ in the workspace, use the command:
```
pm.plot_cohort=true;
```
2. Alternatively, use a parameter setting similar to the command in the previous section:
```
data_all{1} = loadConditionData('/home/Folder/Dual/Movies', {'CLC','AP2'}, {'rfp','egfp'});
dir_alt='/home/AnotherFolder/ForStoring/Dual/DASC_result/';
con_name={'dualControl'};
pm = dasParameter('data_all',data_path,'dir_alt',dir_alt,'con_name',con_name,…
                         'plot_cohort ',true);
```
Note, 'CLC' and 'AP2' are examples of the names of the folders containing the primary and secondary channel movies respectively. 'rfp' and 'egfp' are the fluorescent markers of the two channels.
3. Run DASC in the same way:
```
dasSingleCondition(pm);
```

Epi-TIRF analysis

1. Setup parameters as in **Quick start** while choosing channel number as 2, and select folders specified for each channel. See cmeAnalysis instruction for details. After getting ‘pm’ in the workspace, use the commands:
```
pm.plot_cohort=true;
pm. plot_EpiTIRF=true;
```

2. Alternatively, use the parameter setting as described in the previous section:

   ```
   data_all{1} = loadConditionData('/home/Control/Epi_TIRF/Movies', {'TIRF','Epi'}, {'egfp','egfp'});
   data_all{2} = loadConditionData('/home/siRNAx/Epi_TIRF/Movies', {'TIRF','Epi'}, {'egfp','egfp'});
   dir_alt='/home/AnotherFolder/ForStoring/epi_tirf/DASC_result/';
   con_name={'EpiTIRF_Control', 'EpiTIRF_ siRNAx'};
   pm = dasParameter('data_all',data_path,'dir_alt',dir_alt,'con_name',con_name,…
                                 'plot_cohort ',true, 'plot_ EpiTIRF',true);
   ```

3. To adjust cohort bounds for clathrin coated pits, abortive coats and outlier traces, use respectively:

   ```
   pm.CohortBounds_s_ccp=[5 15];
   pm.CohortBounds_s_ftn=[5 15];
   pm.CohortBounds_s_visitor=[5 15];
   ```

4. Run DASC in the same way:

   ```
   dasSingleCondition(pm);
   ```

Multi-condition (acquired on different days) analysis

1. First, rearrange all the conditions in the following hierarchy data (folder) structure:

   ```
   ../siControl/day1, ../siControl /day2, ../siControl /day3 …
   ../siRNA_1/day1
   ../siRNA_2/day2
   ../siRNA_3/day3
   ../siRNA_4/day3
   …
   ```

   Multi-condition analysis supports only single channel movies. A year-month-day formatted number is required for renaming day1, day2 and so on. For example, day1 is 190512 if the experiment is done on May 12th 2019, and so on.

2. Next, get the path for the structured folder storing all the conditions:

   ```
   dir_movie = '/home/where/the/structured/folder/is';
   ```

3. Next, set a path for where the DASC result of the multi-condition analysis is to be stored:

   ```
   dir_DAS = '/home/where/to/store/DASC_multi/result';
   ```

4. Next, run the command to individually assess each condition:

   ```
   dasMuiltiCondition('dir_movie', dir_movie,'dir_DAS',dir_DAS);
   ```

5. Last, run the command to apply the pooling-bootstrap method for comparing various conditions acquired on different days:

   ```
   dasPoolingBootstrap(dir_DAS, 'I_sub_samp',0.2,'N_samp',10,'N_bs', 20);
   ```

   Here, 'I_sub_samp' is the fraction of traces for normalizing intensities from different days; 'N_samp' is the movie number of each bootstrapped control movie set; 'N_bs' is the number of bootstrap repeats.

Data structure

After running DASC, various folders and '.mat' files will be generated in the result folder defined by the users.

1. 'DAS_all.mat' is the main result file. It is a n by 1 cell array, where n is the number of conditions. The first cell element is always control. For historical reasons, the variable names are not the same as in the article. Each element contains: 'DAS' – d_1 in article; 'DAS_var' – d_2 in article; 'DAS_2' and 'DAS_3' – second and third moment of D series; 'LT' and 'MaxI' – lifetime and maximum intensity of each trace; 'MovieNum' and 'TrackID' – movie and track number of each trace stored in '/ctrl(or treated)/Track_info.mat' (next bullet point).
2. 'Track_info.mat' is the data for intensity traces. The file is a 1 by m cell array, where m is the number of movies. Each element contains: 'Track_id' – track number of each trace; 'A' – intensity matrix.
3. 'DAS_stat_idx.mat' is the statistical result file. It contains: 'idx' – k-medoids classification labels; 'test_p.p_all' – p-values comparing control and treatment conditions in the order of 'CCP rate', 'outlier trace rate', 'abortive coat rate', 'total structure rate', 'CCP%', 'median lifetime' and 'median intensity maxima'.