The Elizabeth R. Miller Brain Observatory (MBO)

LBM-CalmAn-MATLAB Guide

9/25/2024

EXAMPLE DATASET

LBM-CaImAn

- LBM-CaImAn-**MATLAB**
  - 4 step, primarily "batch"
  - Plane-by-plane segmentation
  - Published, vetted
  - Thorough documentation

LBM-CaImAn

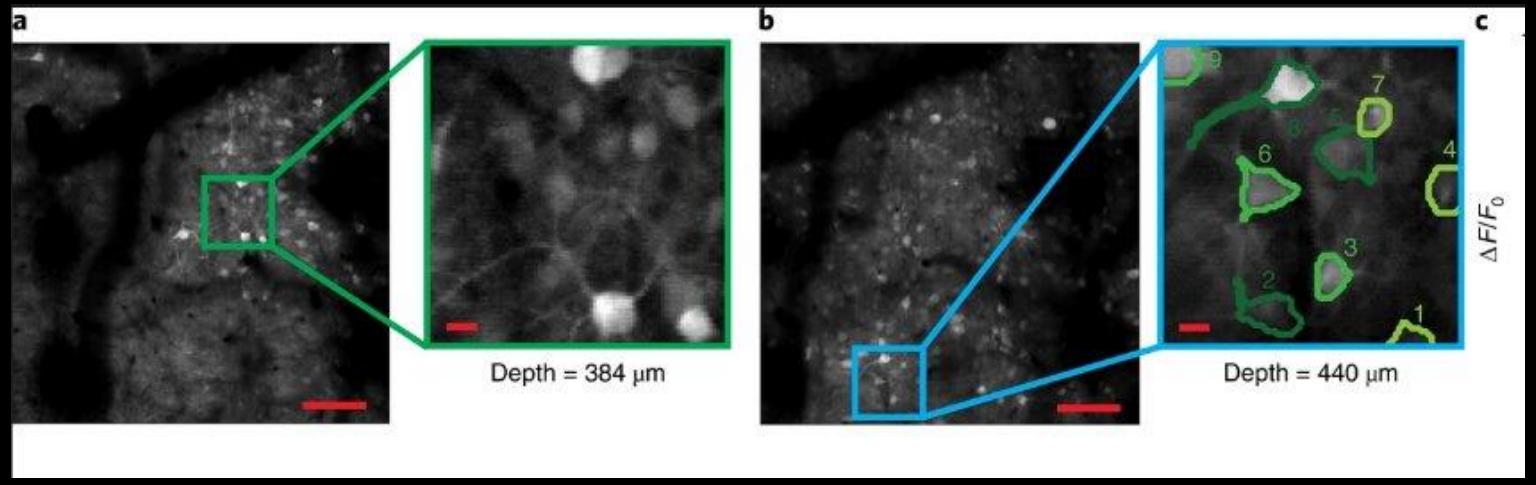
- LBM-CaImAn-**Python**
  - 4 step, batch + interactive
  - Plane-by-plane segmentation
  - Work in progress

nature methods ARTICLES
https://doi.org/10.1038/s41592-021-01239-8

High-speed, cortex-wide volumetric recording of neuroactivity at cellular resolution using light beads microscopy

Jeffrey Demas[1,2], Jason Manley[1,2], Frank Tejera[1], Kevin Barber[1], Hyewon Kim[1], Francisca Martínez Traub[1], Brandon Chen[1] and Alipasha Vaziri[1,2]

Figure 4a-c



We will follow the **same *example dataset*** used throughout the documentation

https://www.nature.com/articles/s41592-021-01239-8

LBM-CaImAn

First Steps
User Guide
Tutorials
API
1. Core Functions
2. Utility Functions
3. Internal Functions
Image Gallery
Glossary
Links

Contents

2.1. Readers
2.2. Writers
2.3. Visualization
2.4. Validation

get_metadata(*filename*)

Extract metadata from a ScanImage TIFF file.

Read and parse Tiff metadata stored in the .tiff header and ScanImage metadata stored in the 'Artist' tag which contains roi sizes/locations and scanning configuration details in a JSON format.

Parameters:

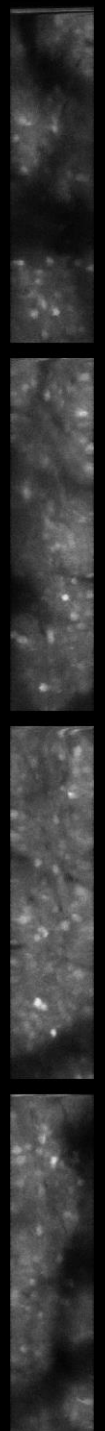
filename : *char*

The full path to the TIFF file from which metadata will be extracted.

Returns:

metadata_out : *struct*

A struct containing metadata such as center and size of the scan field, pixel resolution, image dimensions, number of frames, frame rate, and additional roi data extracted from the TIFF file.



https://millerbrainobservatory.github.io/LBM-CaImAn-MATLAB/api/utility.html

LBM-CaImAn

First Steps
User Guide
Tutorials
API
1. Core Functions
2. Utility Functions
3. Internal Functions
Image Gallery
Glossary
Links

Contents
2.1. Readers
2.2. Writers
2.3. Visualization
2.4. Validation

get_metadata(*filename*)

Extract metadata from a ScanImage TIFF file.

Read and parse Tiff metadata stored in the .tiff header and ScanImage metadata stored in the 'Artist' tag which contains roi sizes/locations and scanning configuration details in a JSON format.

Parameters:

filename : ***char***

The full path to the TIFF file from which metadata will be extracted.

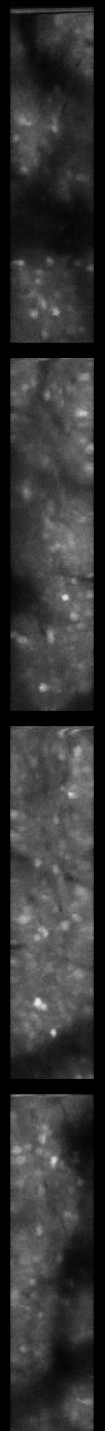
Returns:

metadata_out : ***struct***

A struct containing metadata such as center and size of the scan field, pixel resolution, image dimensions, number of frames, frame rate, and additional roi data extracted from the TIFF file.

```
>> metadata = get_metadata(fullfile(data_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif"))
Elapsed time is 2.317538 seconds.

metadata =

  struct with fields:

                   num_planes: 30
                     num_rois: 4
                   num_frames: 1730
                   frame_rate: 9.6081
                          fov: [600 600]
             pixel_resolution: 1.0208
                sample_format: 'int16'
                 roi_width_px: 144
                roi_height_px: 600
                  tiff_length: 2478
                   tiff_width: 145
                 raw_filename: "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001"
                 raw_filepath: "C:\Users\RBO\caiman_data\animal_01\session_01"
                 raw_fullfile: "C:\Users\RBO\caiman_data\animal_01\session_01\MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif"
    num_lines_between_scanfields: 24
```

https://millerbrainobservatory.github.io/LBM-CaImAn-MATLAB/api/utility.html



LBM-CaImAn

PIPELINE STEPS

1. ***Assembly***
2. ***Motion Correction***
3. ***Segmentation***
4. ***Collation***

https://millerbrainobservatory.github.io/LBM-CaImAn-MATLAB/api/core.html

LBM-CaImAn

PIPELINE STEPS

1. ***Assembly***
2. ***Motion Correction***
3. ***Segmentation***
4. ***Collation***

Core Functions:

1. `convertScanImageTiffToVolume()`
2. `motionCorrectPlane()`
3. `segmentPlane()`
4. `collatePlanes()`

Note to previous users:

- Step 1 and 2 used to be a single step
- Step 4 used to be calculate_offset.m and compare_planes.m

https://millerbrainobservatory.github.io/LBM-CaImAn-MATLAB/api/core.html

LBM-CaImAn

PIPELINE STEPS

1 *Assembly*

2 *Motion Correction*

3 *Segmentation*

4 *Collation*

Core Functions:

1. `convertScanImageTiffToVolume()`
2. `motionCorrectPlane()`
3. `segmentPlane()`
4. `collatePlanes()`

Note to previous users:

- Step 1 and 2 used to be a single step
- Step 4 used to be calculate_offset.m and compare_planes.m

	Raw	Assembled	Registered	Traces / Cell Masks	Traces / Cell Masks (collated)
File Type:	.tiff	.h5	.h5	.h5	.mat
Dimensions:	[x,y,zT]	[x,y,T]*z_i	[x,y,T]*z_i	[K,T]	[K, T]
File Size:	35 Gb	1 Gb * z_i	1 Gb * z_i	3Mb * z_i	10-50Mb
~Time to Complete		4 minutes	~3 hours	~17 hours	~1 hour

z_i *for* i *=1:30*
K = #Neurons

https://millerbrainobservatory.github.io/LBM-CaImAn-MATLAB/api/core.html

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example

```
\repos\LBM-CaImAn-MATLAB\demo_LBM_pipeline.m

%% RUN THIS WITH THE PLAY BUTTON, NOT "RUN SECTION"
% When ran as a script (the "Run" button), this will automatically add the
% core and packages folders to your MATLAB path
% Not needed if the project code is stored in the Documents/MATLAB folder
clc, clear; % !! Careful, this will clear all variables from memory
[fpath, fname, ~] = fileparts(fullfile(mfilename('fullpath'))); % path to this script
addpath(genpath(fullfile(fpath, 'core'))); addpath(genpath(fullfile(fpath, 'packages')));

%% POINT THIS PATH TO WHERE YOUR TIFF FILES LIVE
data_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01');

% To preview metadata for this file without assembling it
% metadata = get_metadata(fullfile(data_path, "filename.tif");
% This happens internally in convertScanImageTiffToVolume()
metadata = get_metadata(fullfile(data_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif"));
disp(metadata);

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);


%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Motion Correction %%%
%%

filename = fullfile(save_path, "assembled_plane_1.h5");
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;

options = NoRMCorreSetParms(...
    'd1', data_size(1),...
    'd2', data_size(2),...
    'grid_size', [64,64], ...
    'bin_width', 100,... % number of frames to avg when updating template
    'max_shift', round(20/metadata.pixel_resolution),... % 20 microns
    'correct_bidir', false... % defaults to true, this was already done in step 1
);
```

https://github.com/MillerBrainObservatory/LBM-CaImAn-MATLAB/blob/master/demo_LBM_pipeline.m

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path

```
\repos\LBM-CaImAn-MATLAB\demo_LBM_pipeline.m

% When ran as a script (the "Run" button), this will automatically add the
% core and packages folders to your MATLAB path
% Not needed if the project code is stored in the Documents/MATLAB folder
clc, clear; % !! Careful, this will clear all variables from memory
[fpath, fname, ~] = fileparts(fullfile(mfilename('fullpath'))); % path to this script
addpath(genpath(fullfile(fpath, 'core'))); addpath(genpath(fullfile(fpath, 'packages')));

%% POINT THIS PATH TO WHERE YOUR TIFF FILES LIVE
data_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01');

% To preview metadata for this file without assembling it
% metadata = get_metadata(fullfile(data_path, "filename.tif");
% This happens internally in convertScanImageTiffToVolume()
metadata = get_metadata(fullfile(data_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif"));
disp(metadata);

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);


%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Motion Correction %%%
%%

filename = fullfile(save_path, "assembled_plane_1.h5");
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;

options = NoRMCorreSetParms(...
    'd1', data_size(1),...
    'd2', data_size(2),...
    'grid_size', [64,64], ...
    'bin_width', 100,... % number of frames to avg when updating template
    'max_shift', round(20/metadata.pixel_resolution),... % 20 microns
    'correct_bidir', false... % defaults to true, this was already done in step 1
);
```

https://github.com/MillerBrainObservatory/LBM-CaImAn-MATLAB/blob/master/demo_LBM_pipeline.m

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path
- Preview session metadata before processing

```
\repos\LBM-CaImAn-MATLAB\demo_LBM_pipeline.m

%% RUN THIS WITH THE PLAY BUTTON, NOT "RUN SECTION"
% When ran as a script (the "Run" button), this will automatically add the
% core and packages folders to your MATLAB path
% Not needed if the project code is stored in the Documents/MATLAB folder
clc, clear; % !! Careful, this will clear all variables from memory
[fpath, fname, ~] = fileparts(fullfile(mfilename('fullpath'))); % path to this script
addpath(genpath(fullfile(fpath, 'core'))); addpath(genpath(fullfile(fpath, 'packages')));

%% POINT THIS PATH TO WHERE YOUR TIFF FILES LIVE
data_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01');

% To preview metadata for this file without assembling it
% metadata = get_metadata(fullfile(data_path, "filename.tif");
% This happens internally in convertScanImageTiffToVolume()
metadata = get_metadata(fullfile(data_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif"));
disp(metadata);

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);


%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Motion Correction %%%
%%

filename = fullfile(save_path, "assembled_plane_1.h5");
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;

options = NoRMCorreSetParms(...
    'd1', data_size(1),...
    'd2', data_size(2),...
    'grid_size', [64,64], ...
    'bin_width', 100,... % number of frames to avg when updating template
    'max_shift', round(20/metadata.pixel_resolution),... % 20 microns
    'correct_bidir', false... % defaults to true, this was already done in step 1
);
```

https://github.com/MillerBrainObservatory/LBM-CaImAn-MATLAB/blob/master/demo_LBM_pipeline.m

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path
- Preview session metadata before processing
- Run pipeline exposing all default values

```
\repos\LBM-CaImAn-MATLAB\demo_LBM_pipeline.m

%% RUN THIS WITH THE PLAY BUTTON, NOT "RUN SECTION"
% When ran as a script (the "Run" button), this will automatically add the
% core and packages folders to your MATLAB path
% Not needed if the project code is stored in the Documents/MATLAB folder
clc, clear; % !! Careful, this will clear all variables from memory
[fpath, fname, ~] = fileparts(fullfile(mfilename('fullpath'))); % path to this script
addpath(genpath(fullfile(fpath, 'core'))); addpath(genpath(fullfile(fpath, 'packages')));

%% POINT THIS PATH TO WHERE YOUR TIFF FILES LIVE
data_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01');

% To preview metadata for this file without assembling it
% metadata = get_metadata(fullfile(data_path, "filename.tif");
% This happens internally in convertScanImageTiffToVolume()
metadata = get_metadata(fullfile(data_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif"));
disp(metadata);

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);


%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Motion Correction %%%
%%

filename = fullfile(save_path, "assembled_plane_1.h5");
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;

options = NoRMCorreSetParms(...
    'd1', data_size(1),...
    'd2', data_size(2),...
    'grid_size', [64,64], ...
    'bin_width', 100,... % number of frames to avg when updating template
    'max_shift', round(20/metadata.pixel_resolution),... % 20 microns
    'correct_bidir', false... % defaults to true, this was already done in step 1
);
```

https://github.com/MillerBrainObservatory/LBM-CaImAn-MATLAB/blob/master/demo_LBM_pipeline.m

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path
- Preview session metadata before processing
- Run pipeline exposing all default values

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);
```

https://github.com/MillerBrainObservatory/LBM-CaImAn-MATLAB/blob/master/demo_LBM_pipeline.m

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path
- Preview session metadata before processing
- Run pipeline exposing all default values
- Only required parameter is the *data_path*

```
%%%
```

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path
- Preview session metadata before processing
- Run pipeline exposing all default values
- Only required parameter is the *data_path*
- *Optional parameters as 'key-value' pairs*

```
required = "path/to/data";
my_function(required, 'key', value);
% OR
my_function(required, ...
    'key', value ...
);
```

demo_LBM_pipeline.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);
```

Required

Optional

Optional + Function-specific

https://github.com/MillerBrainObservatory/LBM-CaImAn-MATLAB/blob/master/demo_LBM_pipeline.m

demo_LBM_pipeline.m

Demo LBM pipeline:

- Full pipeline example
- Automatically sets up package path
- Preview session metadata before processing
- Run pipeline exposing all default values
- Only required parameter is the *data_path*
- *Optional parameters as 'key-value' pairs*

```
required = "path/to/data";
my_function(required, 'key', value);
% OR
my_function(required, ...
    'key', value ...
);
```

```
%%%
```

LBM-CaImAn

PIPELINE STEPS

1 **Assembly** | 2 Motion Correction | 3 Segmentation | 4 Collation

data_path

- should contain **only .tiff files from *this session.***
  - Any other file-types located in this directory will be ignored.
- Output filename will always be assembled_plane_z_i.h5
  - The pipeline relies on the format "pipeline_step_plane_z_i"
- Multi-file sessions are appended *in-order* of filename on disk

```
1. convertScanImageTiffToVolume()
```

Assembled_plane_z_i for i =1:n_planes

LBM-CaImAn

Assembly **Motion Correction** **Segmentation** **Collation**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% Assembly %%%%%%%%%%
%%
% by default, results are saved in the parent directory /function_step
% directory
save_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled\');
convertScanImageTiffToVolume( ...
    data_path, ...
    'save_path', save_path, ...
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'do_figures', 1, ...
    'overwrite', 0, ...
    'fix_scan_phase', 1, ...
    'trim_roi', [0 0 0 0], ... % num pixels to trim for each roi
    'trim_image', [0 0 0 0], ... % num pixels to trim for each roi
    'z_plane_order', [] ... % reshape the planes according to this vector.
    ...                     % num values should match num planes
);
```

Command Window

```
metadata contents:
        num_planes = 30
        num_rois = 4
        num_frames = 1730
        frame_rate = 9.60806
        fov = [600;600]
        pixel_resolution = 1.02083333333333
        sample_format = int16
        num_pixel_xy = [144;600]
        roi_width_px = 144
        roi_height_px = 600
        tiff_length = 2478
        tiff_width = 145
        raw_filename = MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001
        raw_filepath = C:\Users\RBO\caiman_data\animal_01\session_01
        raw_fullfile = C:\Users\RBO\caiman_data\animal_01\session_01\MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif
        num_lines_between_scanfields = 24
```

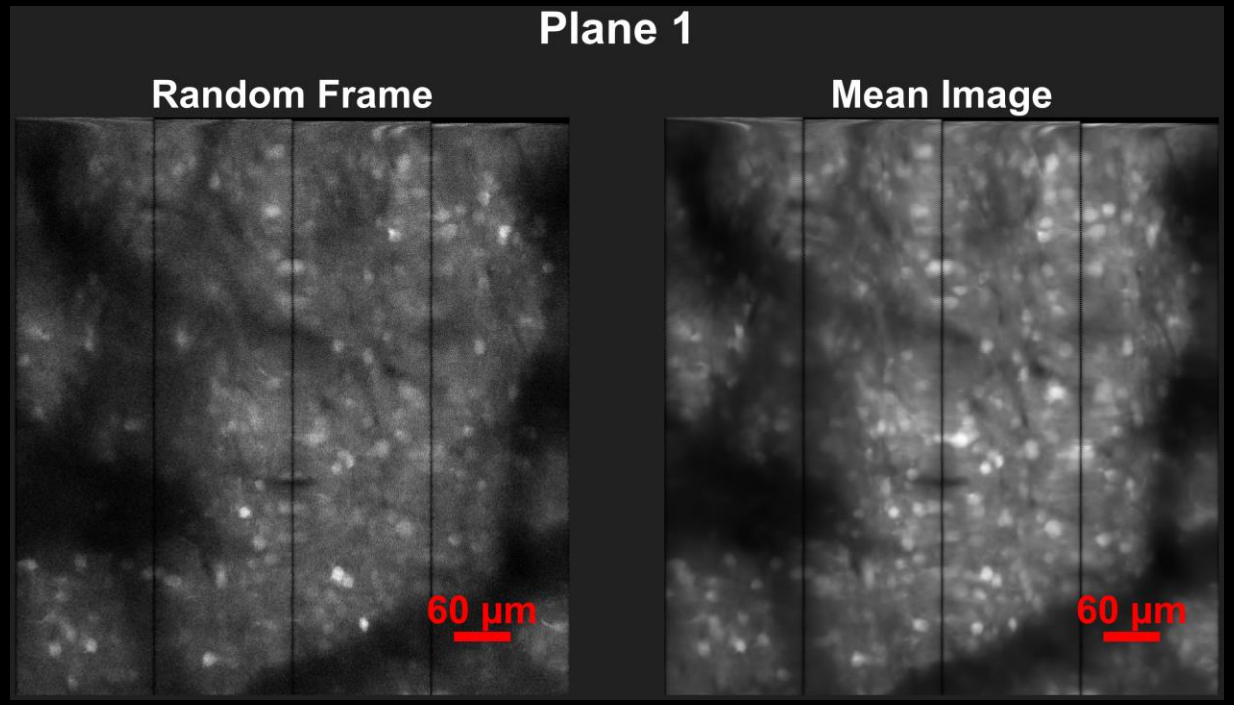
caiman_data › animal_01 › assembled_default ›

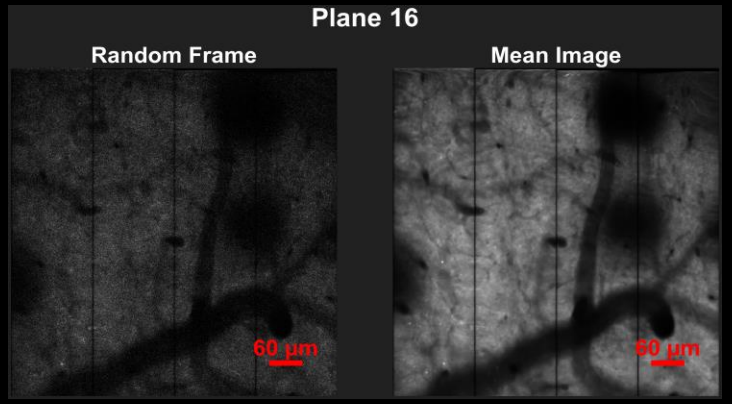
Name	Date modified	Type	Size
figures	9/19/2024 10:52 AM	File folder	
assembled_plane_1.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_2.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_3.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_4.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_5.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_6.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_7.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_8.h5	9/19/2024 10:49 AM	H5 File	1,170,504 KB
assembled_plane_9.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_10.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_11.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_12.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_13.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_14.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_15.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_16.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_17.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_18.h5	9/19/2024 10:50 AM	H5 File	1,170,504 KB
assembled_plane_19.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_20.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_21.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_22.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_23.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_24.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_25.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_26.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_27.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_28.h5	9/19/2024 10:51 AM	H5 File	1,170,504 KB
assembled_plane_29.h5	9/19/2024 10:52 AM	H5 File	1,170,504 KB
assembled_plane_30.h5	9/19/2024 10:52 AM	H5 File	1,170,504 KB

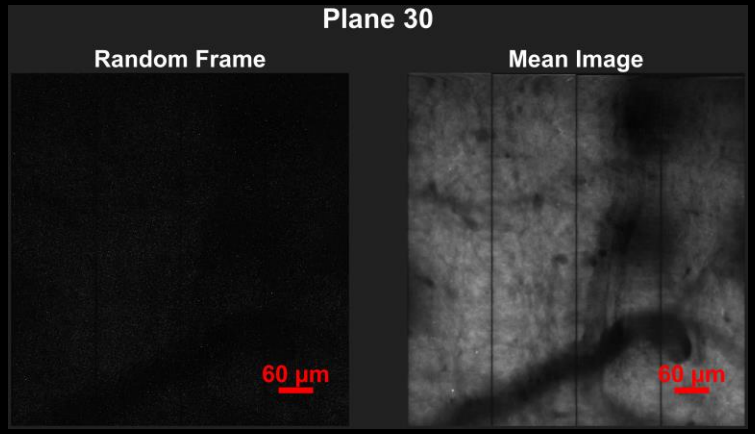
LBM-CaImAn

Assembly *Motion Correction* *Segmentation* *Collation*

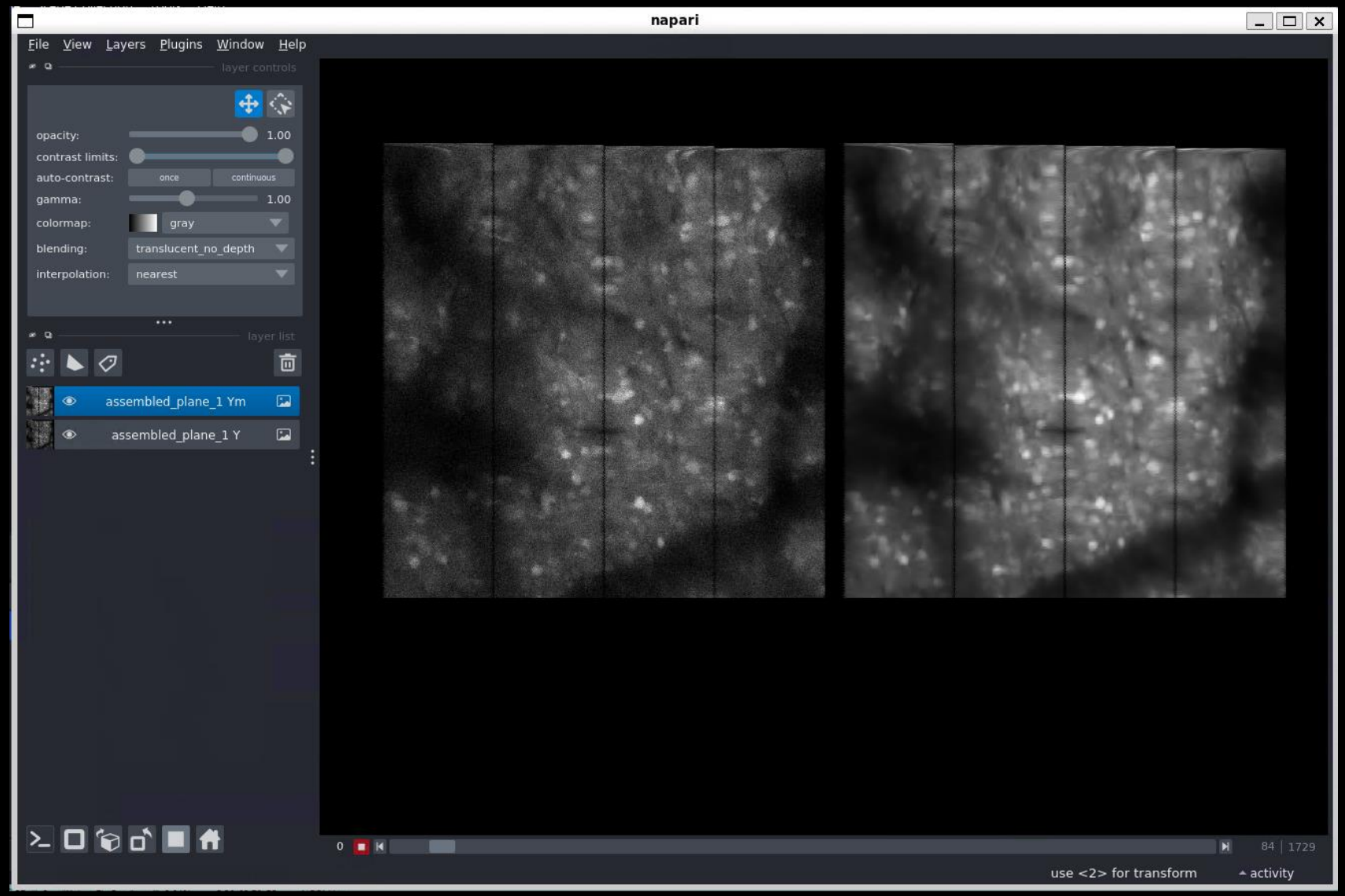
Name	Date
plane_1.fig	9/19/2024 11:46 PM
plane_1.png	9/19/2024 11:46 PM
plane_2.fig	9/19/2024 11:46 PM
plane_2.png	9/19/2024 11:46 PM
plane_3.fig	9/19/2024 11:47 PM
plane_3.png	9/19/2024 11:47 PM
plane_4.fig	9/19/2024 11:47 PM
plane_4.png	9/19/2024 11:47 PM
plane_5.fig	9/19/2024 11:47 PM
plane_5.png	9/19/2024 11:47 PM
plane_6.fig	9/19/2024 11:47 PM
plane_6.png	9/19/2024 11:47 PM
plane_7.fig	9/19/2024 11:47 PM
plane_7.png	9/19/2024 11:47 PM
plane_8.fig	9/19/2024 11:47 PM
plane_8.png	9/19/2024 11:47 PM
plane_9.fig	9/19/2024 11:47 PM
plane_9.png	9/19/2024 11:47 PM
plane_10.fig	9/19/2024 11:47 PM
plane_10.png	9/19/2024 11:47 PM
plane_11.fig	9/19/2024 11:47 PM
plane_11.png	9/19/2024 11:47 PM
plane_12.fig	9/19/2024 11:47 PM
plane_12.png	9/19/2024 11:47 PM
plane_13.fig	9/19/2024 11:48 PM
plane_13.png	9/19/2024 11:48 PM
plane_14.fig	9/19/2024 11:48 PM
plane_14.png	9/19/2024 11:48 PM
plane_15.fig	9/19/2024 11:48 PM
plane_15.png	9/19/2024 11:48 PM
plane_16.fig	9/19/2024 11:48 PM
plane_16.png	9/19/2024 11:48 PM
plane_17.fig	9/19/2024 11:48 PM
plane_17.png	9/19/2024 11:48 PM







All .h5 files can be opened in napari



https://www.napari-hub.org/plugins/napari-h5

Function-Specific Parameters:

fix_scan_phase : ***logical, optional***

Whether to correct for bi-directional scan artifacts. Default is true.

trim_roi : ***double, optional***

Pixels to trim from [left, right, top, bottom] edge of each **INDIVIDUAL ROI** before horizontally concatenating the ROI's within an image. Default is [0 0 0 0]. Only applies to ScanImage multi-ROI recordings.

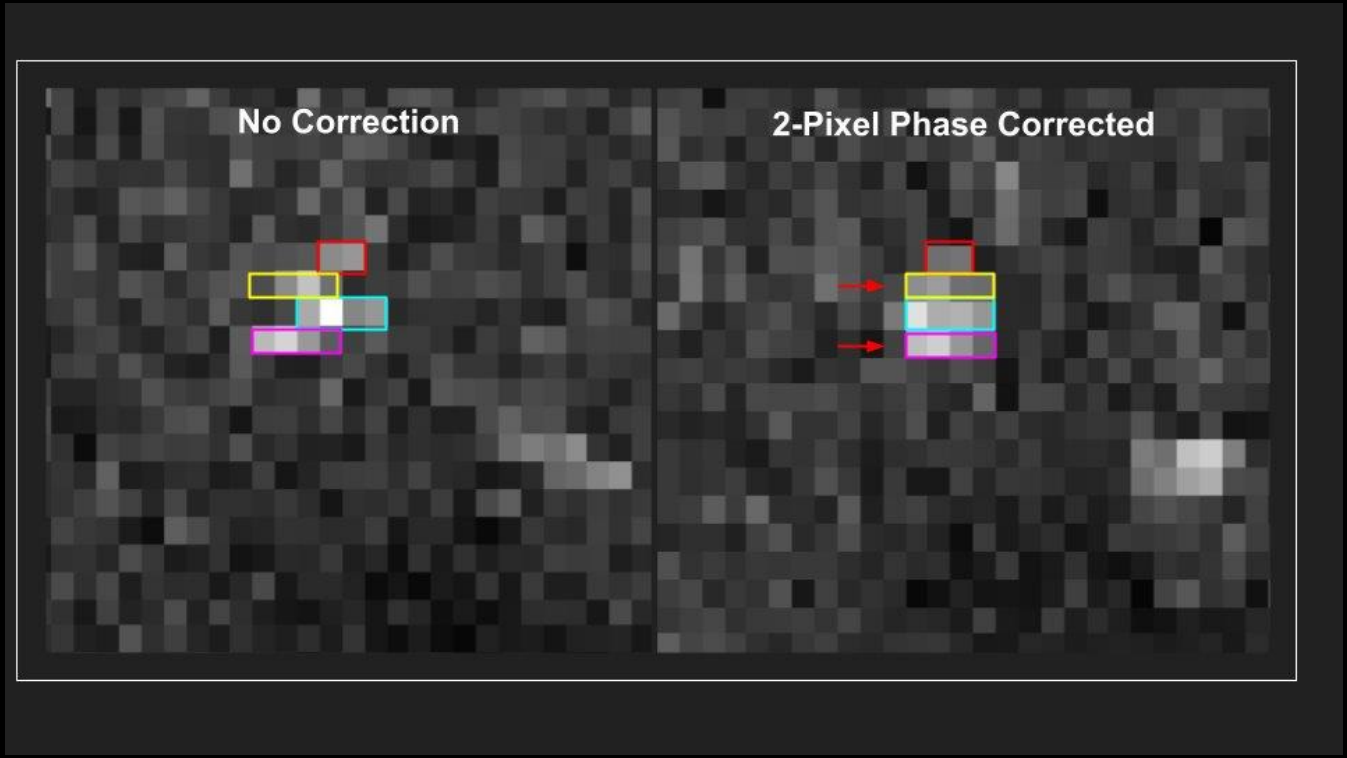
trim_image : ***double, optional***

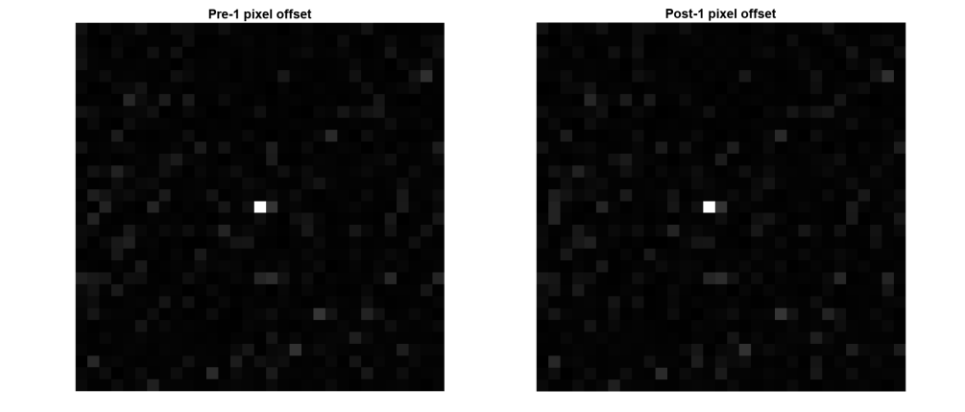
Pixels to trim from [left, right, top, bottom] edge of each **TILED IMAGE** before horizontally concatenating the ROI's within an image. Default is [0 0 0 0]. Only applies to ScanImage multi-ROI recordings.

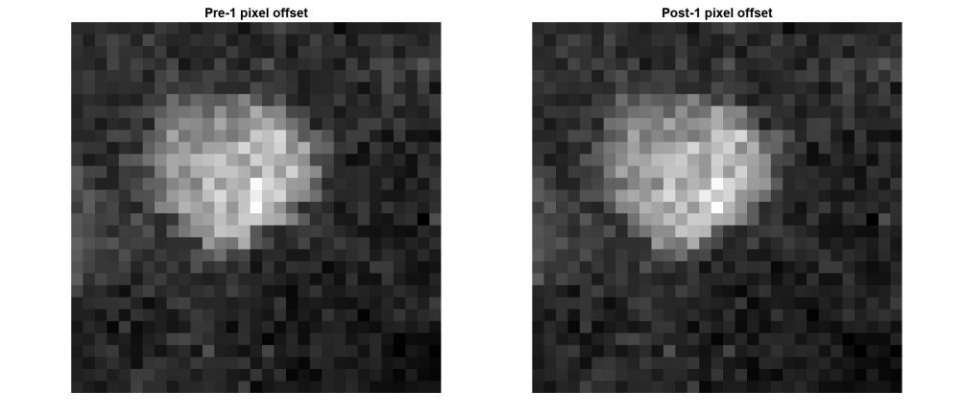
z_plane_order : ***double, optional***

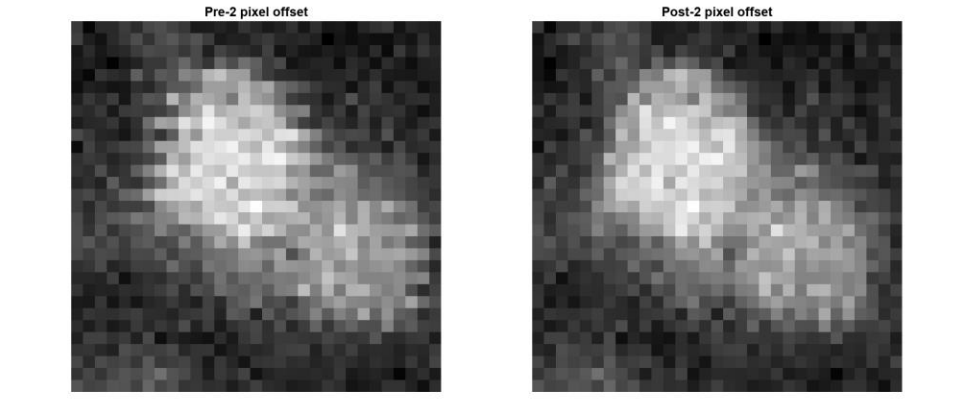
If interleaved z-planes are not ordered from 1-n_planes, reorder the stack using this array as index.

fix_scan_phase



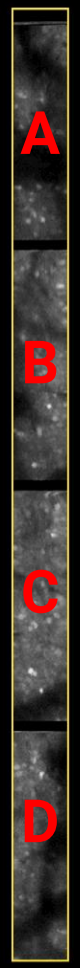




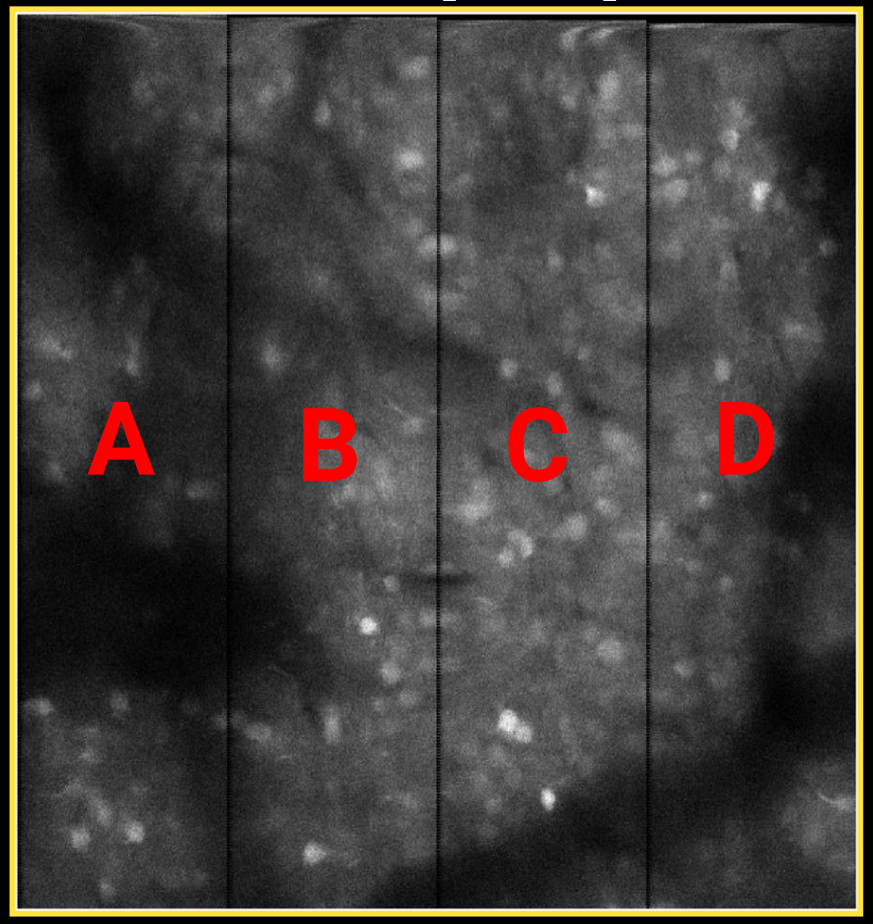


trim_roi

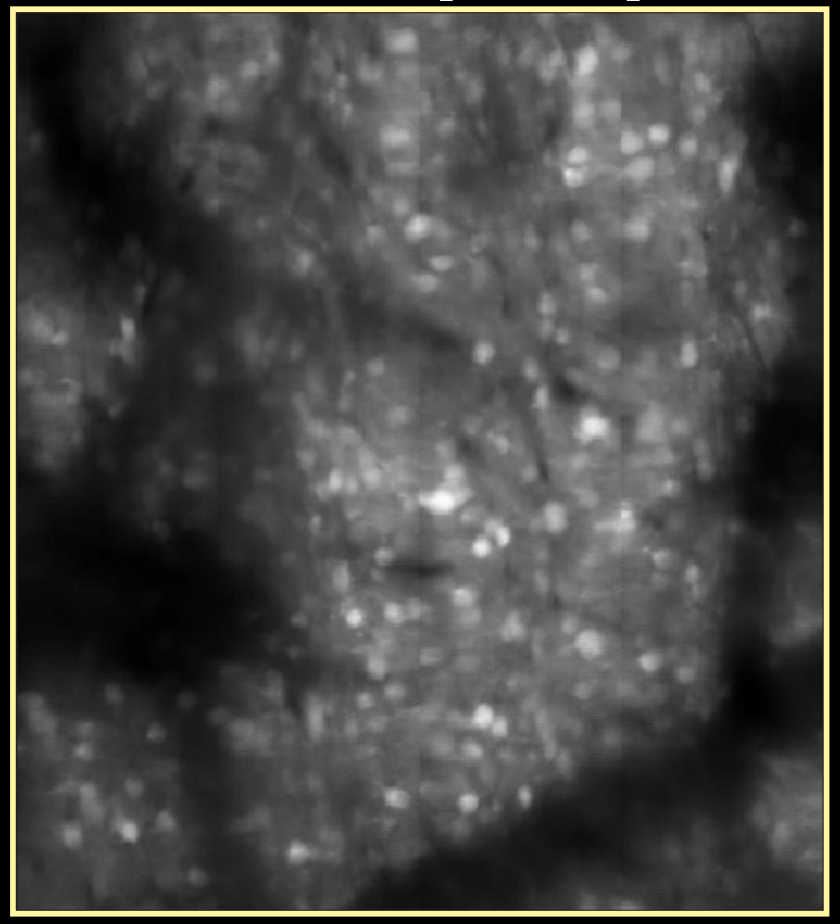
Raw TIFF Page



trim_roi = [0,0,0,0]

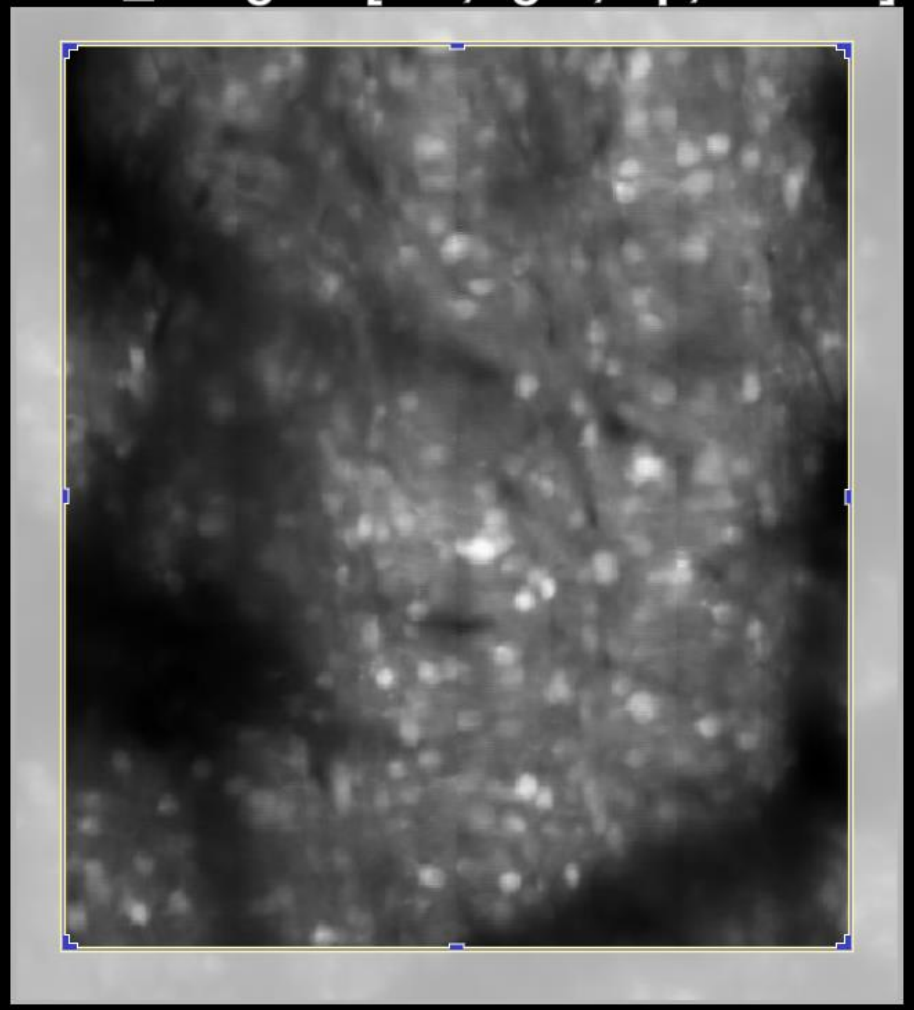


trim_roi = [6,6,17,0]



trim_image

trim_image = [left,right,top,bottom]



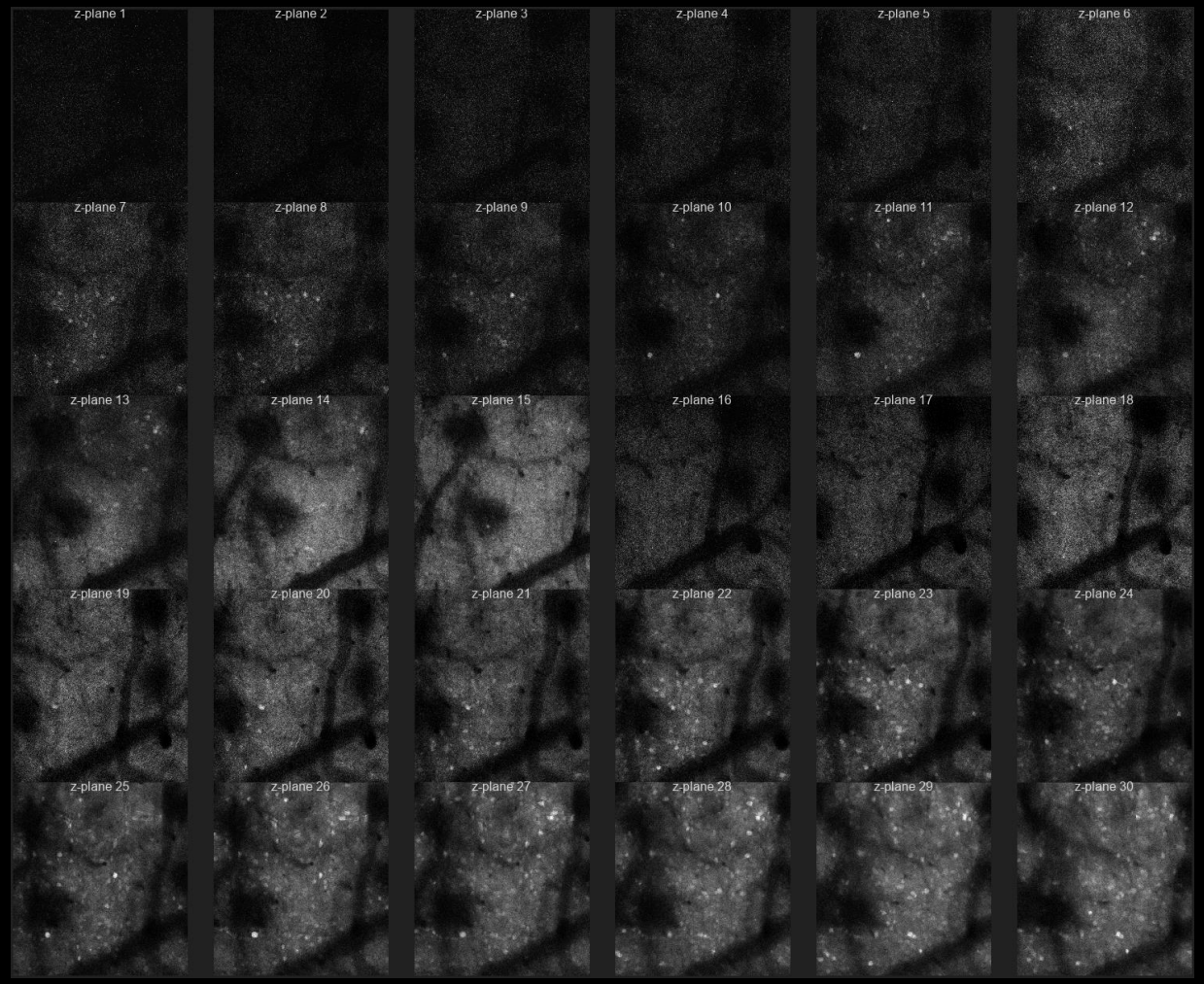
If you want to trim pixels *without* affecting inter-ROI pixels

z_plane_order

- Vector containing integers for z-plane ordering

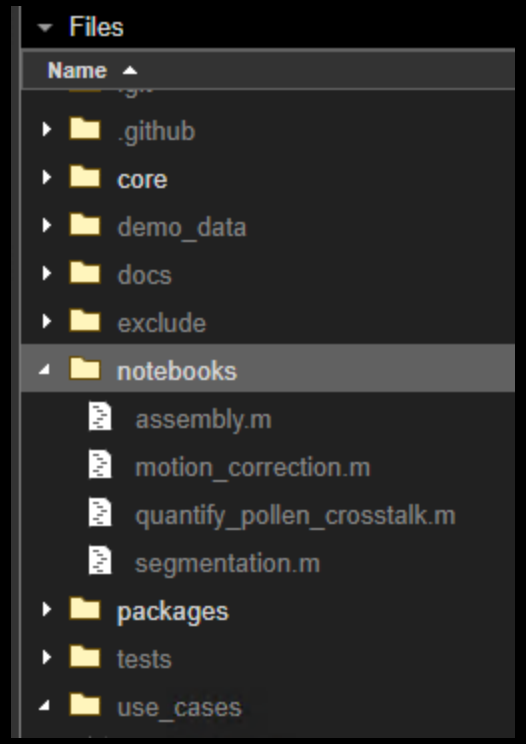
- Changes the order of each z-plane
- To flip plane 1 and 2:
  - z_plane_order = [2 1 3 4 5 6 .... n_planes]

z_plane_order



To create these summary images:
notebooks/assembly.m

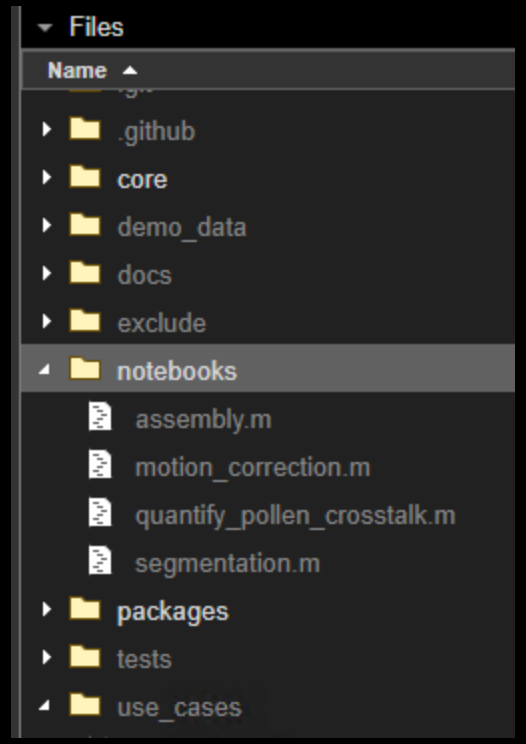
Assembly Notebook



C:\Users\RBO\repos\LBM-CaImAn-MATLAB\notebooks\assembly.m

```
% base TIFF directory
tiff_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\');

% where your assembled files are saved
assembly_path = fullfile(tiff_path, 'assembled');
% where you want to save the outputs
save_path = fullfile(tiff_path, 'outputs');

% Ensure the save_path directory exists; create it if it doesn't
if ~exist(save_path, 'dir')
    mkdir(save_path);
end

% Use the TIFF path to get the metadata file
metadata_file = fullfile(tiff_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif");
metadata = get_metadata(metadata_file);

%% After assembly, collect a single frame from each assembled image
num_planes = metadata.num_planes;

% Load and normalize images
for plane_idx = 1:num_planes
    % Construct the filename using assembly_path and plane index
    filename = fullfile(assembly_path, sprintf('assembled_plane_%d.h5', plane_idx));
    if isfile(filename)
        Y = h5read(filename, '/Y');
        if plane_idx == 1
            sizeY = size(Y);
            frames = zeros(sizeY(1), sizeY(2), num_planes);
        end
        frames(:, :, plane_idx) = Y(:, :, 2);  % Store each mean image
    end
end

% Normalize the images to the range [0, 255]
frames = uint8(255 * mat2gray(frames));
```

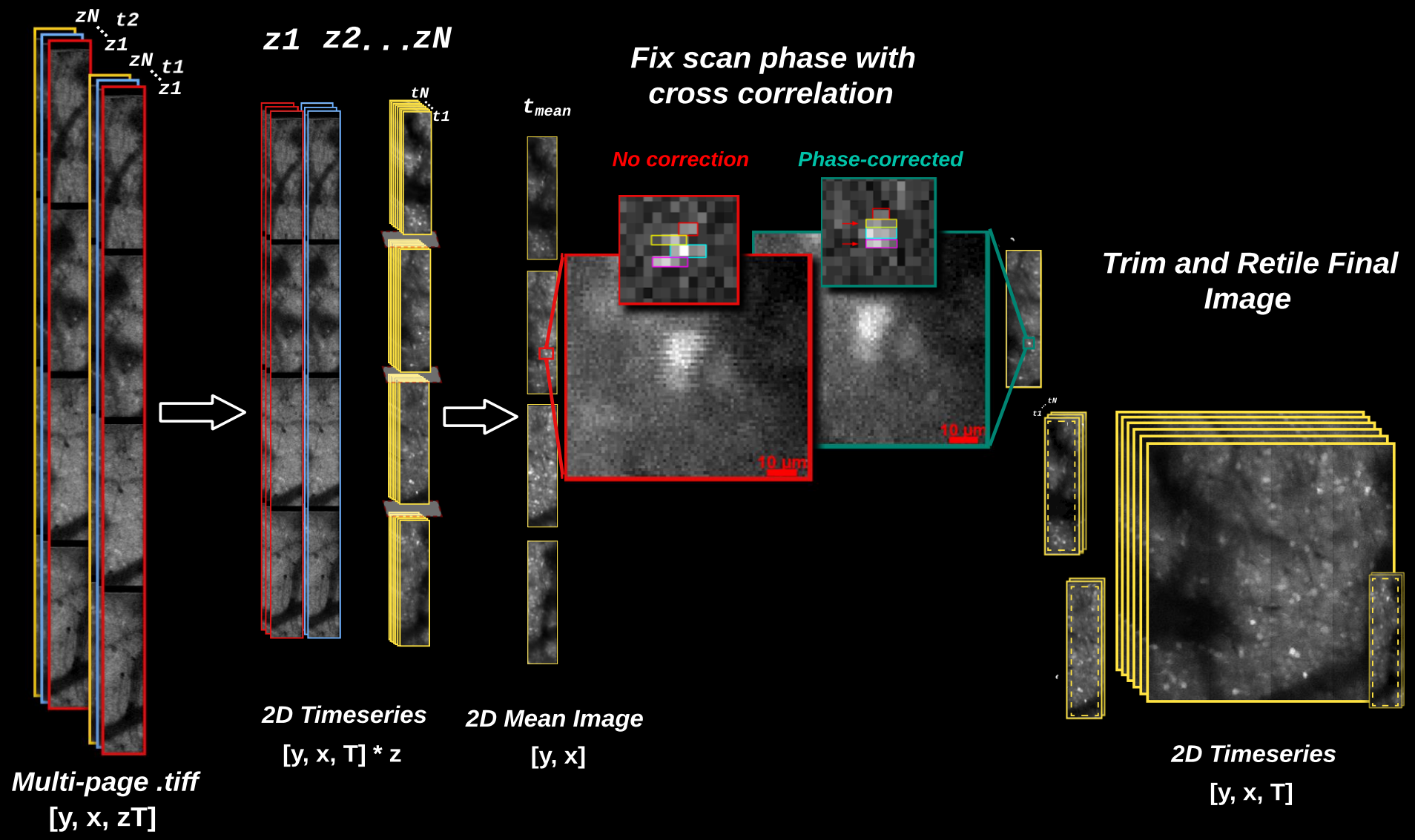
Assembly Notebook



User input

```
% base TIFF directory
tiff_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\');

% where your assembled files are saved
assembly_path = fullfile(tiff_path, 'assembled');
% where you want to save the outputs
save_path = fullfile(tiff_path, 'outputs');

% Ensure the save_path directory exists; create it if it doesn't
if ~exist(save_path, 'dir')
    mkdir(save_path);
end

% Use the TIFF path to get the metadata file
metadata_file = fullfile(tiff_path, "MH70_0p6mm_FOV_50_550um_depth_som_stim_199mW_3min_M1_00001_00001.tif");
metadata = get_metadata(metadata_file);

%% After assembly, collect a single frame from each assembled image
num_planes = metadata.num_planes;

% Load and normalize images
for plane_idx = 1:num_planes
    % Construct the filename using assembly_path and plane index
    filename = fullfile(assembly_path, sprintf('assembled_plane_%d.h5', plane_idx));
    if isfile(filename)
        Y = h5read(filename, '/Y');
        if plane_idx == 1
            sizeY = size(Y);
            frames = zeros(sizeY(1), sizeY(2), num_planes);
        end
        frames(:, :, plane_idx) = Y(:, :, 2);  % Store each mean image
    end
end

% Normalize the images to the range [0, 255]
frames = uint8(255 * mat2gray(frames));
```

Step 1. Assembly



PIPELINE STEPS

1 Assembly | **2 Motion Correction** | 3 Segmentation | 4 Collation

data_path

- Points to save_path used in convertScanImageTiffToVolume()
  - Checks for the presence of assembled_plane_Z_i
- Output filename will always be motion_corrected_plane_z_i.h5

```
2. motionCorrectPlane()
```

planez_i for i =start_plane:end_plane

Function-Specific Parameters:

num_cores : ***double, integer, positive***

Number of cores to use for computation. The value is limited to a maximum of 24 cores.

start_plane : ***double, integer, positive***

The starting plane index for processing.

end_plane : ***double, integer, positive***

The ending plane index for processing. Must be greater than or equal to start_plane.

options : ***struct***

NormCorre Params Object,

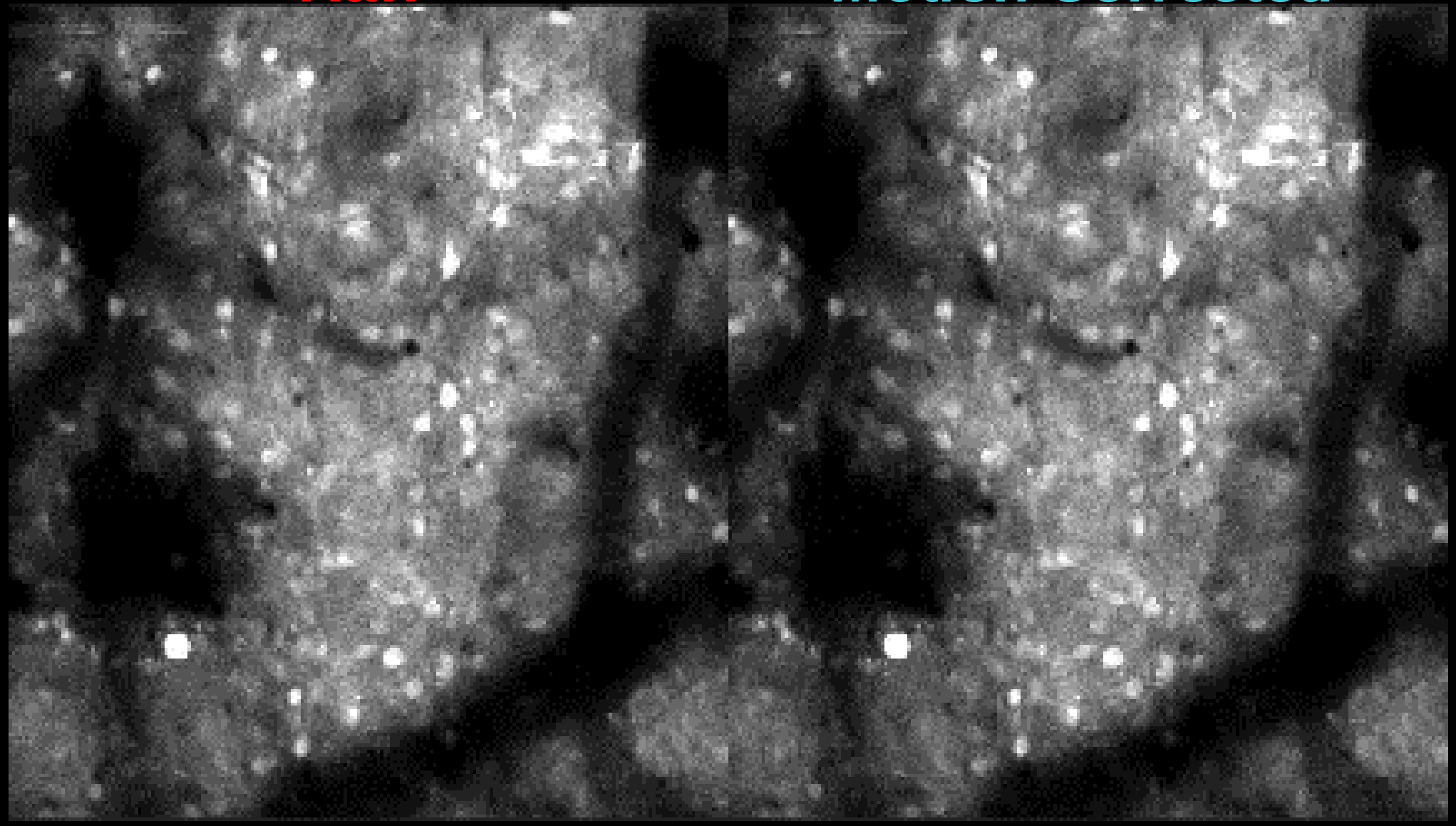
demo_LBM_pipeline.m

```
%%%
```

LBM-CaImAn

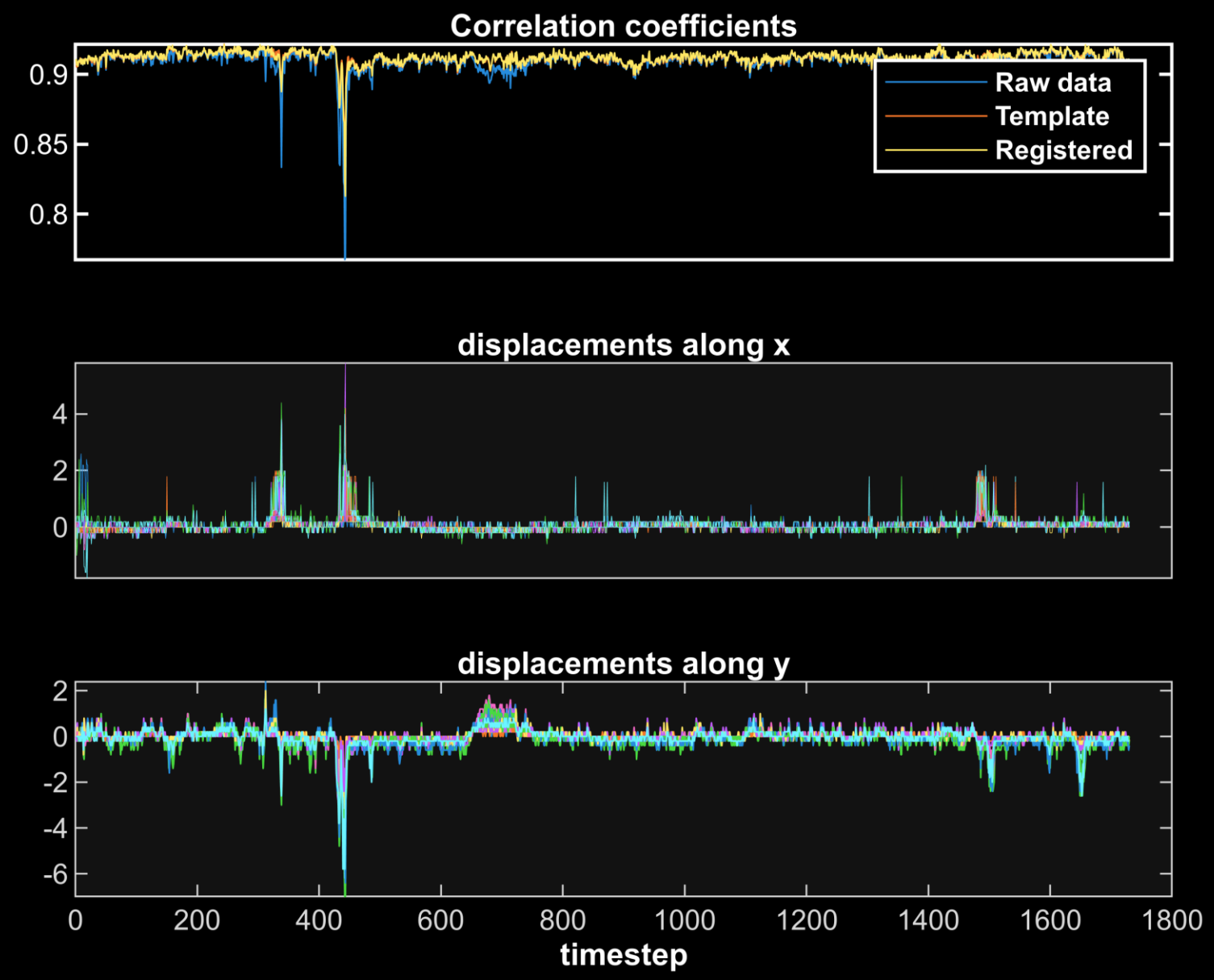
Assembly Motion Correction Segmentation Collation

Raw Motion Corrected



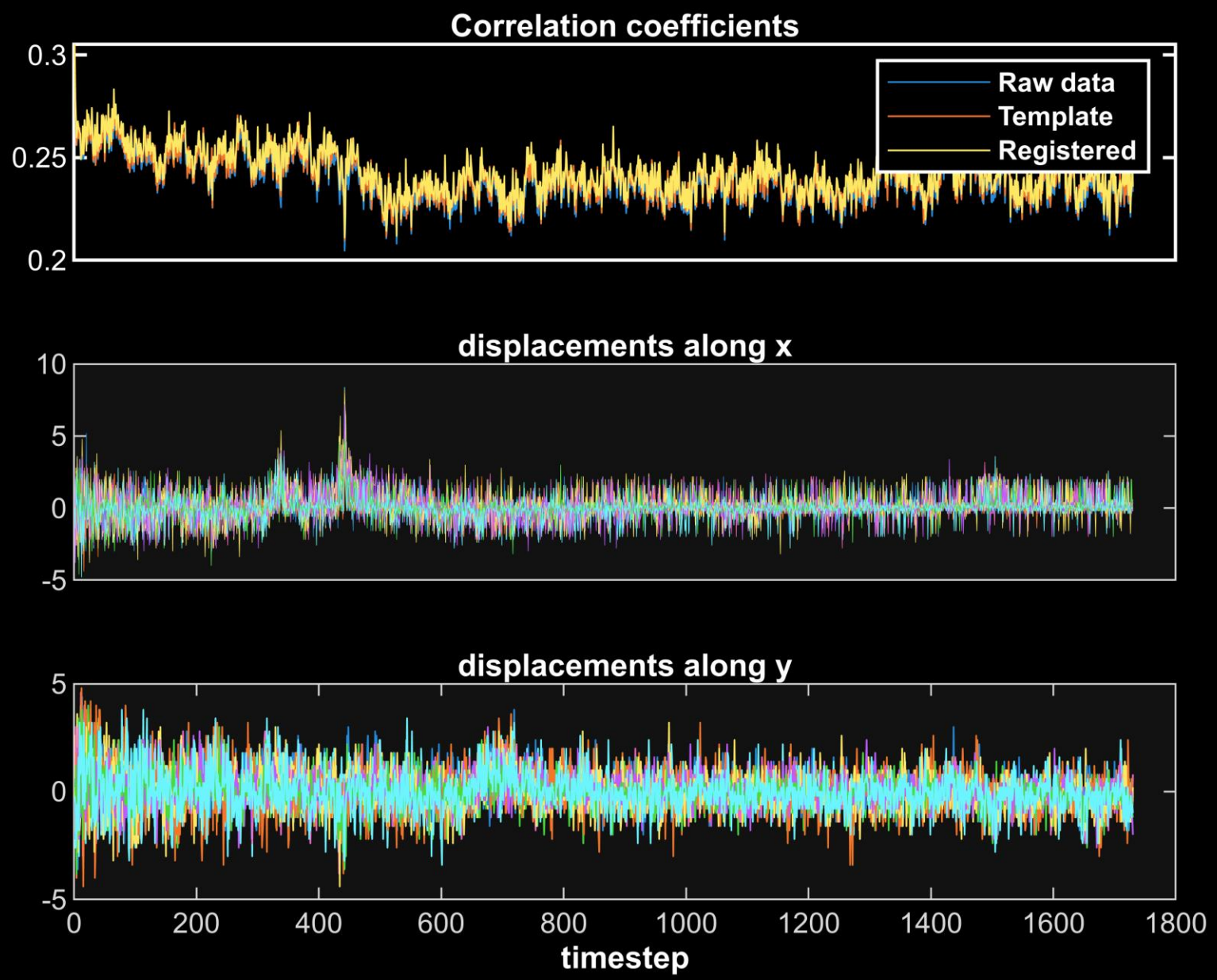
plane_1_metrics.fig
plane_1_metrics.png
plane_1_shifts.fig
plane_1_shifts.png
plane_2_metrics.fig
plane_2_metrics.png
plane_2_shifts.fig
plane_2_shifts.png
plane_3_metrics.fig
plane_3_metrics.png
plane_3_shifts.fig
plane_3_shifts.png
plane_4_metrics.fig
plane_4_metrics.png
plane_4_shifts.fig
plane_4_shifts.png
plane_5_metrics.fig
plane_5_metrics.png
plane_5_shifts.fig
plane_5_shifts.png
plane_6_metrics.fig
plane_6_metrics.png
plane_6_shifts.fig
plane_6_shifts.png
plane_7_metrics.fig
plane_7_metrics.png
plane_7_shifts.fig
plane_7_shifts.png
plane_8_metrics.fig
plane_8_metrics.png
plane_8_shifts.fig
plane_8_shifts.png

plane_26_shifts.png



plane_1_metrics.fig
plane_1_metrics.png
plane_1_shifts.fig
plane_1_shifts.png
plane_2_metrics.fig
plane_2_metrics.png
plane_2_shifts.fig
plane_2_shifts.png
plane_3_metrics.fig
plane_3_metrics.png
plane_3_shifts.fig
plane_3_shifts.png
plane_4_metrics.fig
plane_4_metrics.png
plane_4_shifts.fig
plane_4_shifts.png
plane_5_metrics.fig
plane_5_metrics.png
plane_5_shifts.fig
plane_5_shifts.png
plane_6_metrics.fig
plane_6_metrics.png
plane_6_shifts.fig
plane_6_shifts.png
plane_7_metrics.fig
plane_7_metrics.png
plane_7_shifts.fig
plane_7_shifts.png
plane_8_metrics.fig
plane_8_metrics.png
plane_8_shifts.fig
plane_8_shifts.png

plane_1_shifts.png

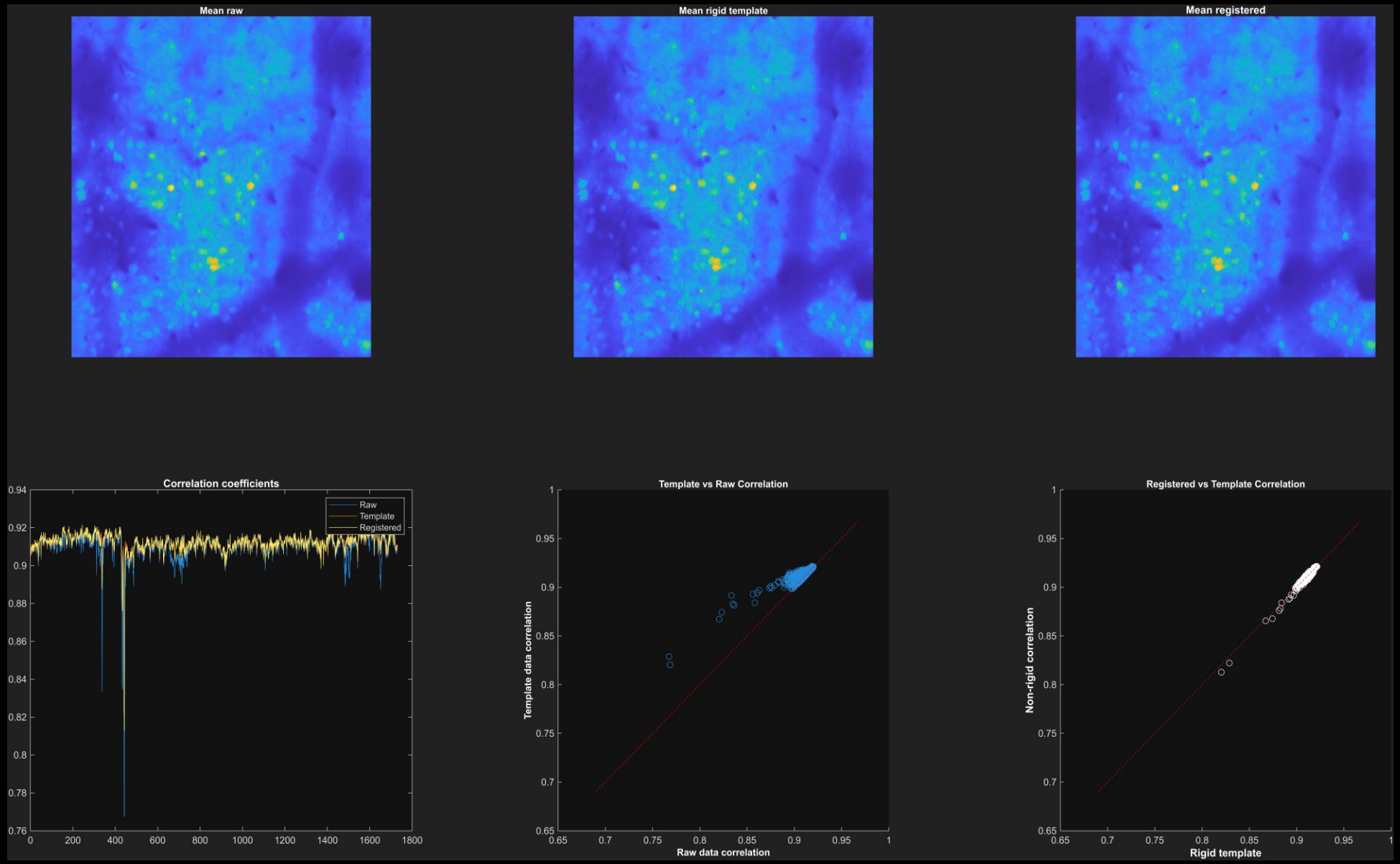


LBM-CaImAn

Assembly Motion Correction Segmentation Collation

plane_26_metrics.png

- plane_1_metrics.fig
- plane_1_metrics.png
- plane_1_shifts.fig
- plane_1_shifts.png
- plane_2_metrics.fig
- plane_2_metrics.png
- plane_2_shifts.fig
- plane_2_shifts.png
- plane_3_metrics.fig
- plane_3_metrics.png
- plane_3_shifts.fig
- plane_3_shifts.png
- plane_4_metrics.fig
- plane_4_metrics.png
- plane_4_shifts.fig
- plane_4_shifts.png
- plane_5_metrics.fig
- plane_5_metrics.png
- plane_5_shifts.fig
- plane_5_shifts.png
- plane_6_metrics.fig
- plane_6_metrics.png
- plane_6_shifts.fig
- plane_6_shifts.png
- plane_7_metrics.fig
- plane_7_metrics.png
- plane_7_shifts.fig
- plane_7_shifts.png
- plane_8_metrics.fig
- plane_8_metrics.png
- plane_8_shifts.fig
- plane_8_shifts.png

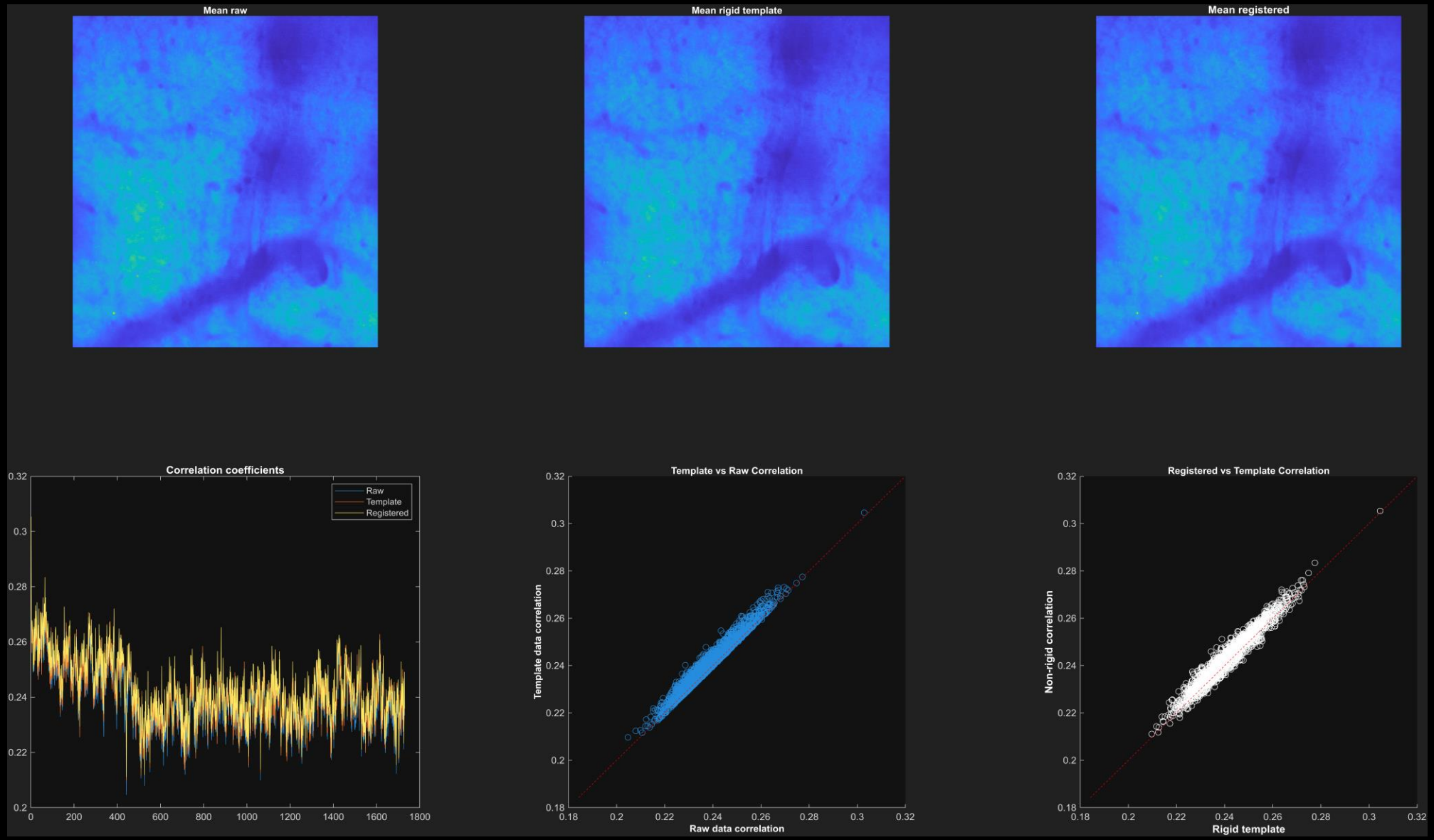


LBM-CaImAn

Assembly Motion Correction Segmentation Collation

plane_1_metrics.png

- plane_1_metrics.fig
- plane_1_metrics.png
- plane_1_shifts.fig
- plane_1_shifts.png
- plane_2_metrics.fig
- plane_2_metrics.png
- plane_2_shifts.fig
- plane_2_shifts.png
- plane_3_metrics.fig
- plane_3_metrics.png
- plane_3_shifts.fig
- plane_3_shifts.png
- plane_4_metrics.fig
- plane_4_metrics.png
- plane_4_shifts.fig
- plane_4_shifts.png
- plane_5_metrics.fig
- plane_5_metrics.png
- plane_5_shifts.fig
- plane_5_shifts.png
- plane_6_metrics.fig
- plane_6_metrics.png
- plane_6_shifts.fig
- plane_6_shifts.png
- plane_7_metrics.fig
- plane_7_metrics.png
- plane_7_shifts.fig
- plane_7_shifts.png
- plane_8_metrics.fig
- plane_8_metrics.png
- plane_8_shifts.fig
- plane_8_shifts.png



notebooks/motion_correction.m

```
%% Motion Correction Grid Search

%% POINT THIS PATH TO YOUR DESIRED PLANE, ASSEMBLED FILES, AND REGISTERED FILES
plane = 26;

assembled_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled');
motion_corrected_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\motion_corrected');

%%

filename = fullfile(assembled_path, sprintf("assembled_plane_%d.h5", plane));
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;
% data_size = h5info(filename, '/Y').ChunkSize(1:2); % you can get size from h5 as well

% Put any variables you want to search inside [], separated by a comma
shifts_um = [5, 20, 40];
grid_sizes = [64, 128];

for i=1:numel(shifts_um)
    disp(i);
    disp(size(shifts_um))
    max_shift = shifts_um(i);
    for j=1:numel(grid_sizes)
        grid_size = grid_sizes(j);

        % use variables to save the file
        grid_search_savepath = sprintf( ...
                "%s/plane_%d_max_shift_%d_grid_size_%dx%d", ...
                plane,assembled_path,max_shift,grid_size(1), grid_size(1) ...
        );

        options = NoRMCorreSetParms(...
            'd1', data_size(1),...
            'd2', data_size(2),...
            'grid_size', grid_size, ...
            'bin_width', 100,... % number of frames to avg when updating template
            'max_shift', max_shift,...
            'correct_bidir', false... % defaults to true, this was already done in step 1
        );

        motionCorrectPlane( ...
            assembled_path, ... % we used this to save extracted data
            'save_path', grid_search_savepath, ... % save registered data here
            'ds', '/Y', ... % where we saved the last step in h5
            'debug_flag', 0, ...
            'overwrite', 1, ...
            'num_cores', 23, ...
            'start_plane', plane, ...
            'end_plane', plane,  ...
            'options', options ...
        );
    end
end
```

Motion Correction Grid Search Template

notebooks/motion_correction.m

Motion Correction Grid Search Template

```
%% Motion Correction Grid Search

%% POINT THIS PATH TO YOUR DESIRED PLANE, ASSEMBLED FILES, AND REGISTERED FILES
plane = 26;

assembled_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled');
motion_corrected_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\motion_corrected');

%%

filename = fullfile(assembled_path, sprintf("assembled_plane_%d.h5", plane));
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;
% data_size = h5info(filename, '/Y').ChunkSize(1:2); % you can get size from h5 as well

% Put any variables you want to search inside [], separated by a comma
shifts_um = [5, 20, 40];
grid_sizes = [64, 128];

for i=1:numel(shifts_um)
    disp(i);
    disp(size(shifts_um))
    max_shift = shifts_um(i);
    for j=1:numel(grid_sizes)
        grid_size = grid_sizes(j);

        % use variables to save the file
        grid_search_savepath = sprintf( ...
                "%s/plane_%d_max_shift_%d_grid_size_%dx%d", ...
                plane,assembled_path,max_shift,grid_size(1), grid_size(1) ...
        );

        options = NoRMCorreSetParms(...
            'd1', data_size(1),...
            'd2', data_size(2),...
            'grid_size', grid_size, ...
            'bin_width', 100,... % number of frames to avg when updating template
            'max_shift', max_shift,...
            'correct_bidir', false... % defaults to true, this was already done in step 1
        );

        motionCorrectPlane( ...
            assembled_path, ... % we used this to save extracted data
            'save_path', grid_search_savepath, ... % save registered data here
            'ds', '/Y', ... % where we saved the last step in h5
            'debug_flag', 0, ...
            'overwrite', 1, ...
            'num_cores', 23, ...
            'start_plane', plane, ...
            'end_plane', plane,  ...
            'options', options ...
        );
    end
end
```

Define a list of parameter values to test

notebooks/motion_correction.m

Motion Correction Grid Search Template

```
%% Motion Correction Grid Search

%% POINT THIS PATH TO YOUR DESIRED PLANE, ASSEMBLED FILES, AND REGISTERED FILES
plane = 26;

assembled_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled');
motion_corrected_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\motion_corrected');

%%

filename = fullfile(assembled_path, sprintf("assembled_plane_%d.h5", plane));
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;
% data_size = h5info(filename, '/Y').ChunkSize(1:2); % you can get size from h5 as well

% Put any variables you want to search inside [], separated by a comma
shifts_um = [5, 20, 40];
grid_sizes = [64, 128];

for i=1:numel(shifts_um)
    disp(i);
    disp(size(shifts_um))
    max_shift = shifts_um(i);
    for j=1:numel(grid_sizes)
        grid_size = grid_sizes(j);

        % use variables to save the file
        grid_search_savepath = sprintf( ...
                "%s/plane_%d_max_shift_%d_grid_size_%dx%d", ...
                plane,assembled_path,max_shift,grid_size(1), grid_size(1) ...
        );

        options = NoRMCorreSetParms(...
            'd1', data_size(1),...
            'd2', data_size(2),...
            'grid_size', grid_size, ...
            'bin_width', 100,... % number of frames to avg when updating template
            'max_shift', max_shift,...
            'correct_bidir', false... % defaults to true, this was already done in step 1
        );

        motionCorrectPlane( ...
            assembled_path, ... % we used this to save extracted data
            'save_path', grid_search_savepath, ... % save registered data here
            'ds', '/Y', ... % where we saved the last step in h5
            'debug_flag', 0, ...
            'overwrite', 1, ...
            'num_cores', 23, ...
            'start_plane', plane, ...
            'end_plane', plane,  ...
            'options', options ...
        );
    end
end
```

Define a list of parameter values to test

For i=1:numel(parameter1)
For j=1:numel(parameter2)
...
For all parameters.

notebooks/motion_correction.m

Motion Correction Grid Search Template

```
%% Motion Correction Grid Search

%% POINT THIS PATH TO YOUR DESIRED PLANE, ASSEMBLED FILES, AND REGISTERED FILES
plane = 26;

assembled_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled');
motion_corrected_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\motion_corrected');

%%

filename = fullfile(assembled_path, sprintf("assembled_plane_%d.h5", plane));
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;
% data_size = h5info(filename, '/Y').ChunkSize(1:2); % you can get size from h5 as well

% Put any variables you want to search inside [], separated by a comma
shifts_um = [5, 20, 40];
grid_sizes = [64, 128];

for i=1:numel(shifts_um)
    disp(i);
    disp(size(shifts_um))
    max_shift = shifts_um(i);
    for j=1:numel(grid_sizes)
        grid_size = grid_sizes(j);

        % use variables to save the file
        grid_search_savepath = sprintf( ...
                "%s/plane_%d_max_shift_%d_grid_size_%dx%d", ...
                plane,assembled_path,max_shift,grid_size(1), grid_size(1) ...
        );

        options = NoRMCorreSetParms(...
            'd1', data_size(1),...
            'd2', data_size(2),...
            'grid_size', grid_size, ...
            'bin_width', 100,... % number of frames to avg when updating template
            'max_shift', max_shift,...
            'correct_bidir', false... % defaults to true, this was already done in step 1
        );

        motionCorrectPlane( ...
            assembled_path, ... % we used this to save extracted data
            'save_path', grid_search_savepath, ... % save registered data here
            'ds', '/Y', ... % where we saved the last step in h5
            'debug_flag', 0, ...
            'overwrite', 1, ...
            'num_cores', 23, ...
            'start_plane', plane, ...
            'end_plane', plane,  ...
            'options', options ...
        );
    end
end
```

Define a list of parameter values to test

For i=1:numel(parameter1)
For j=1:numel(parameter2)
...
For all parameters.

View and save files to mp4

```
%% Play registration movie alongside raw movie

motion_corrected_filename = fullfile(motion_corrected_path, sprintf("motion_corrected_plane_%d.h5", plane));
raw_movie = h5read(raw_filename, '/Y');
motion_corrected_data = h5read(raw_filename, '/Y');

play_movie({raw_movie, motion_corrected_data})

%% Save registration movie alongside raw movie

write_frames_to_mp4(raw_movie, motion_corrected_path, metadata.frame_rate);
```

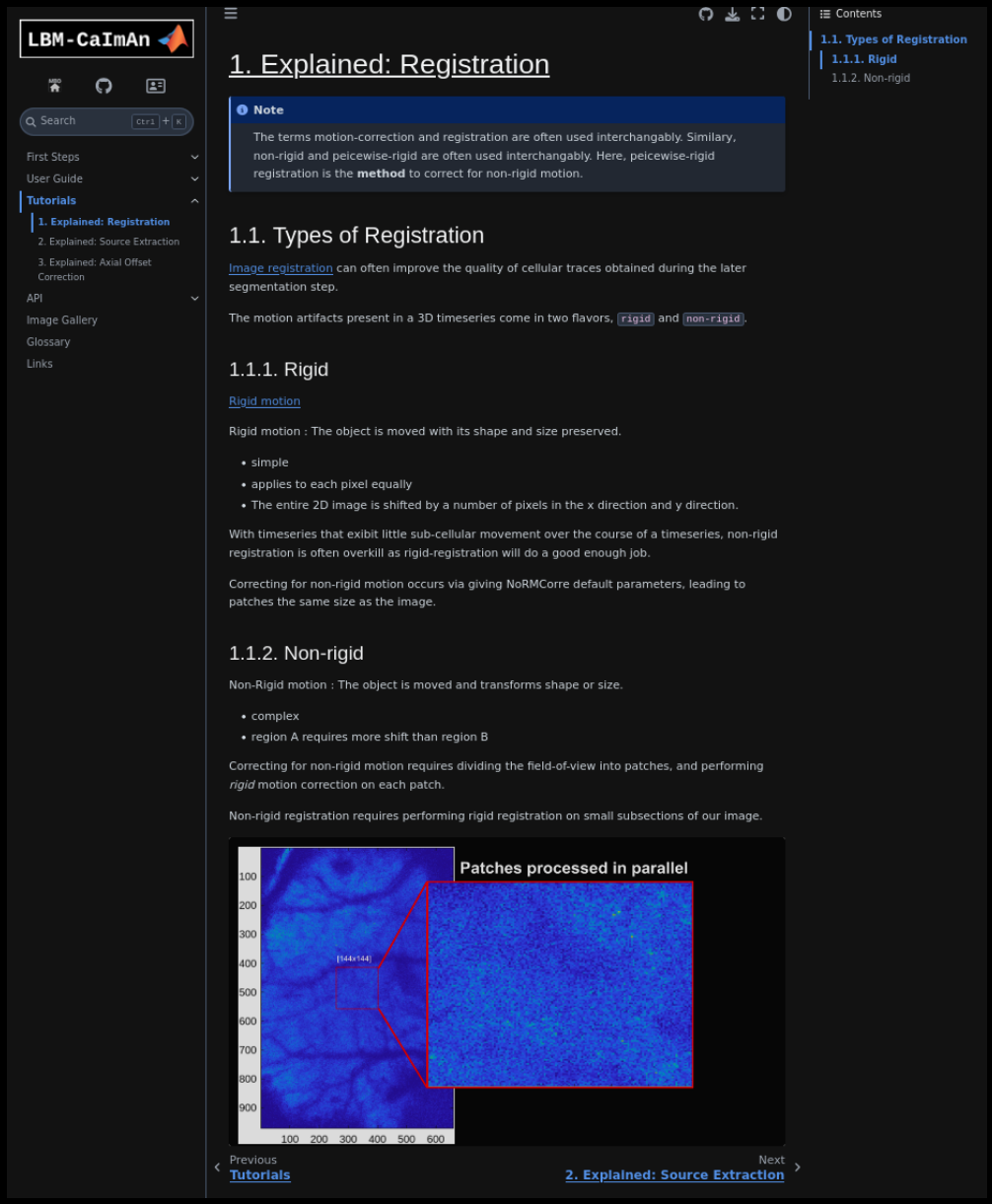
notebooks/motion_correction.m

```
%% Motion Correction Grid Search

%% POINT THIS PATH TO YOUR DESIRED PLANE, ASSEMBLED FILES, AND REGISTERED FILES
plane = 26;

assembled_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\assembled');
motion_corrected_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\motion_corrected');

%%

filename = fullfile(assembled_path, sprintf("assembled_plane_%d.h5", plane));
metadata = read_h5_metadata(filename, '/Y');

data_size = metadata.data_size;
% data_size = h5info(filename, '/Y').ChunkSize(1:2); % you can get size from h5 as well

% Put any variables you want to search inside [], separated by a comma
shifts_um = [5, 20, 40];
grid_sizes = [64, 128];

for i=1:numel(shifts_um)
    disp(i);
    disp(size(shifts_um))
    max_shift = shifts_um(i);
    for j=1:numel(grid_sizes)
        grid_size = grid_sizes(j);

        % use variables to save the file
        grid_search_savepath = sprintf( ...
                "%s/plane_%d_max_shift_%d_grid_size_%dx%d", ...
                plane,assembled_path,max_shift,grid_size(1), grid_size(1) ...
        );

        options = NoRMCorreSetParms(...
            'd1', data_size(1),...
            'd2', data_size(2),...
            'grid_size', grid_size, ...
            'bin_width', 100,... % number of frames to avg when updating template
            'max_shift', max_shift,...
            'correct_bidir', false... % defaults to true, this was already done in step 1
        );

        motionCorrectPlane( ...
            assembled_path, ... % we used this to save extracted data
            'save_path', grid_search_savepath, ... % save registered data here
            'ds', '/Y', ... % where we saved the last step in h5
            'debug_flag', 0, ...
            'overwrite', 1, ...
            'num_cores', 23, ...
            'start_plane', plane, ...
            'end_plane', plane,  ...
            'options', options ...
        );
    end
end
```

max_shift_5_grid_size_64x64	9/19/2024 2:55 PM	File folder
max_shift_5_grid_size_128x128	9/19/2024 2:58 PM	File folder
max_shift_20_grid_size_64x64	9/19/2024 3:01 PM	File folder
max_shift_20_grid_size_128x128	9/19/2024 3:03 PM	File folder
max_shift_40_grid_size_64x64	9/19/2024 3:06 PM	File folder
max_shift_40_grid_size_128x128	9/19/2024 3:08 PM	File folder

Tutorials and guides for each step in the docs

. . . on to step 3: segmentation.



https://millerbrainobservatory.github.io/LBM-CaImAn-MATLAB/tutorials/registration_tutorial.html

PIPELINE STEPS

1 Assembly | 2 Motion Correction | **3 Segmentation** | 4 Collation

data_path

- Points to save_path used in motionCorrectPlane()
  - Checks for the presence of motion_corrected_plane_Z_i
- Output filename will always be segmented_plane_z_i.h5

3. segmentPlane()

planez_i for i =start_plane:end_plane

Function-Specific Parameters:

num_cores : ***double, integer, positive***

Number of cores to use for computation. The value is limited to a maximum of 24 cores.

start_plane : ***double, integer, positive***

The starting plane index for processing.

end_plane : ***double, integer, positive***

The ending plane index for processing. Must be greater than or equal to start_plane.

options : ***struct***

key:value pairs of all of your CNMF parameters. See the example parameters in the LBM_demo_pipeline.

demo_LBM_pipeline.m

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples

tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics

mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);

% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                         % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and decay)
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);

segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

demo_LBM_pipeline.m

Pull metadata using the first z-plane

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples

tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics

mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);

% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                        % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and decay)
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);

segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

demo_LBM_pipeline.m

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples
```

Pull metadata using the first z-plane

```
tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics
```

Primary CNMF setting variables

```
mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);

% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                         % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and decay)
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);

segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

demo_LBM_pipeline.m

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples
```

Pull metadata using the first z-plane

```
tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics
```

Primary CNMF setting variables

```
mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);
```

Use these settings to setup patches, grids, and expected neuron sizes

```
% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);

% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                        % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and decay)
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);

segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

demo_LBM_pipeline.m

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples
```

Pull metadata using the first z-plane

```
tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics
```

Primary CNMF setting variables

```
mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);
```

Use these settings to setup patches, grids, and expected neuron sizes

```
% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);
```

Base K on the # of neurons you expect **PER PATCH**

```
% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                        % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and decay)
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);

segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

demo_LBM_pipeline.m

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples
```

Pull metadata using the first z-plane

```
tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics
```

Primary CNMF setting variables

```
mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);
```

Use these settings to setup patches, grids, and expected neuron sizes

```
% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);
```

Base K on the # of neurons you expect **PER PATCH**

```
% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                        % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and deca
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);
```

Use above variables to set CNMF params just like NoRMCorre

```
segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

demo_LBM_pipeline.m

```
%% 3) CNMF Plane-by-plane Segmentation

% Pull metadata from a single file
filename = fullfile(data_path, 'motion_corrected/', "motion_corrected_plane_1.h5");

% Gather relevnt metadata
metadata = read_h5_metadata(filename, '/Y');
data_size = h5info(filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples
```

Pull metadata using the first z-plane

```
tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % threshold for merging
min_SNR = 1.4;                      % 1.4 is liberal,
space_thresh = 0.2;                 % threhsold for selection of neurons by space
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics
```

Primary CNMF setting variables

```
mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
patch_size = round(650/pixel_resolution).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);
```

Use these settings to setup patches, grids, and expected neuron sizes

```
% number of components based on assumption of 9.2e4 neurons/mm^3
K = ceil(9.2e4.*20e-9.*(pixel_resolution.*patch_size(1)).^2);
```

Base K on the # of neurons you expect **PER PATCH**

```
% Set caiman parameters
 options = CNMFSetParms(...
    'd1',data_size(1),'d2',data_size(2),...                        % dimensionality of the FOV
    'deconv_method','constrained_foopsi',...    % neural activity deconvolution method
    'temporal_iter',3,...                       % number of block-coordinate descent steps
    'maxIter',15,...                            % number of NMF iterations during initialization
    'spatial_method','regularized',...          % method for updating spatial components
    'df_prctile',20,...                         % take the median of background fluorescence to compute baseline fluorescence
    'p',p,...                                   % order of autoregressive system (p = 0 no dynamics, p=1 just decay, p = 2, both rise and deca
    'gSig',tau,...                              % half size of neuron
    'merge_thr',merge_thresh,...                % how correlated 2 neurons must be to merge
    'nb',1,...                                  % number of background components (keep at 1 or 2, 2 for more noisy background)
    'gnb',3,...
    'min_SNR',min_SNR,...                       % minimum SNR threshold
    'space_thresh',space_thresh ,...            % space correlation threshold
    'decay_time',0.5,...                        % decay time of transients, GCaMP6s
    'size_thr', sz, ...
    'min_size', round(tau), ...                 % minimum size of ellipse axis (default: 3)
    'max_size', 2*round(tau), ....              % maximum size of ellipse axis (default: 8)
    'max_size_thr',mx,...                       % maximum size of each component in pixels (default: 300)
    'min_size_thr',mn,...                       % minimum size of each component in pixels (default: 9)
    'rolling_length',ceil(frame_rate*5),...
    'fr', frame_rate ...
);
```

Use above variables to set CNMF params just like NoRMCorre

```
segmentPlane( ...
    fullfile(data_path, 'motion_corrected/'), ... % we used this to save extracted data
    'dataset_name', '/Y', ... % where we saved the last step in h5 (default)
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'num_cores', 23, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'options', options ...
);
```

Call segmentPlane() from 1 to num_planes with 23 cores

```
% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % how temporally correlated any two neurons need to be to be merged
min_SNR = 1.4;                      % signal-noise ratio needed to accept this neuron as initialized (1.4 is liberal)
space_thresh = 0.2;                 % how spatially correlated nearby px need to be to be considered for segmentation
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics
```

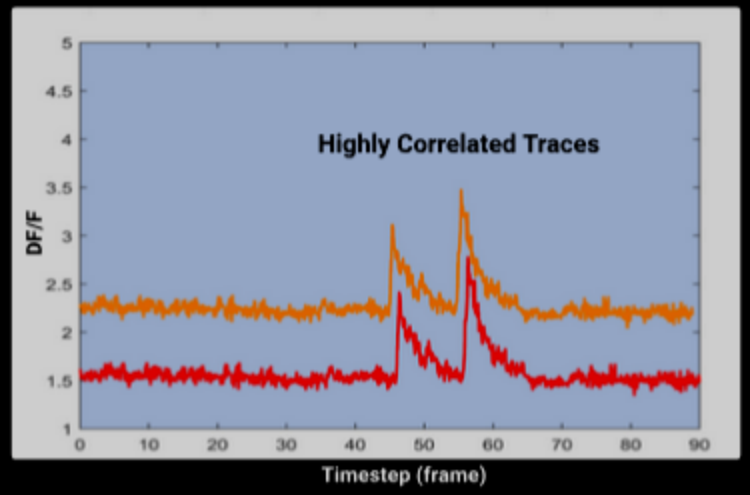
```
% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;          % how temporally correlated any two neurons need to be to be merged
min_SNR = 1.4;               % signal-noise ratio needed to accept this neuron as initialized (1.4 is liberal)
space_thresh = 0.2;          % how spatially correlated nearby px need to be to be considered for segmentation
sz = 0.1;                    % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                       % order of dynamics
```

Most influential CNMF parameters/variables in documentation

3.3.1. `merge_thresh`

A correlation coefficient determining the amount of correlation between pixels in time needed to consider two neurons the same neuron.

- The lower your resolution, the more "difficult" it is for CNMF to distinguish between two tight neurons, thus use a lower merge threshold.
- This parameter heavily effects the number of neurons processed. It's always better to have to many neurons vs too few, as you can never get a lost neuron back, but you can invalidate neurons in post-processing.



3.3.2. `min_SNR`

The minimum signal-noise ratio needed for the algorithm to accept a neurons initialization (accept it as valid).

This value is used for an event exceptionality test, which tests the probabilty if some "exceptional events" (like a spike).

> **Hint**
>
> **If this value is low, even a very slight deviation in signal will be considered exceptional and many background-neurons will be accepted.**

- The likeihood of observing the actual trace value over N samples given an estimated noise distribution.
- The function first estimates the noise distribution by considering the dispersion around the mode.

This probability is used to order the components according to "most likely to be exceptional".

3.3.3. `Tau`

Half-size of your neurons.

- Tau is the `half-size` of a neuron. If a neuron is $10\mu m$, `Tau` will be a $5\mu m$.
- In general, round up.
- This changes depending on the area of the brain you're in and should be adjusted to match the ~cell size of the brain region.

> **Note**
>
> If this parameter is not included, they will be calculated for you based on the `pixel_resolution`, frame rate and image size in the metadata.

- For example, `Tau` is a widely talked about parameter being the half-size of your neuron.

This is calculated by default as `7.5/pixel_resolution`.

This makes sense if we assume an ~neuron size of $14\mu m$.

3.3.4. `P`

This is the autoregressive order of the system.

- It is a measure of how the signal changes with respect to time.
- This value will always be 1 or 2, depending on the frame rate of the video and the dynamics of the calcium indicator.

> **Tip**
>
> In general, **If your indicator takes >1 frame to rise/decay, P=2 (slow)** otherwise, P=1 (fast)

notebooks/segmentation.m

```
%% Interactive Segmentation

%% POINT THIS TO YOUR DESIRED PLANE and MOTION-CORRECTED FILEPATH
plane = 26;
motion_corrected_path = fullfile('C:\Users\RBO\caiman_data\animal_01\session_01\motion_corrected');

%%

motion_corrected_filename = fullfile(motion_corrected_path, sprintf("motion_corrected_plane_%d.h5", plane));
metadata = read_h5_metadata(motion_corrected_filename, '/Y');

%%
% data_size = metadata.data_size;
data_size = h5info(motion_corrected_filename, '/Y').ChunkSize(1:2); % you can get size from h5 as well


%%
% Pull metadata from a single file

% Gather relevent metadata
metadata = read_h5_metadata(motion_corrected_filename, '/Y');
data_size = h5info(motion_corrected_filename, '/Y').ChunkSize(1:2);
pixel_resolution = metadata.pixel_resolution;
frame_rate = metadata.frame_rate;
d = data_size(1)*data_size(2);      % total number of samples

tau = ceil(7.5./pixel_resolution);  % (a little smaller than) half-size of neuron

% USER-SET VARIABLES - CHANGE THESE TO IMPROVE SEGMENTATION

merge_thresh = 0.8;                 % how temporally correlated any two neurons need to be to be merged
min_SNR = 1.4;                      % signal-noise ratio needed to accept this neuron as initialized (1.4 is liberal)
space_thresh = 0.2;                 % how spatially correlated nearby px need to be to be considered for segmentation
sz = 0.1;                           % If masks are too large, increase sz, if too small, decrease. 0.1 lowest.
p = 2;                              % order of dynamics

mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up; basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
fov = metadata.fov(1);
patch_size = round(fov).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% Visualize Patches

figure;
img = h5read(motion_corrected_filename, '/Ym');

imagesc(img); axis image; axis off;
hold on;

for k = 1:length(patches)
    patch = patches{k};
    x = [patch(3), patch(4), patch(4), patch(3)];
    y = [patch(1), patch(1), patch(2), patch(2)];
    fill(x, y, 'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none');  % Green with 50% opacity
end
```

Similar setup to motion_correction.m notebook

Pull metadata using the first motion_corrected_plane_Z

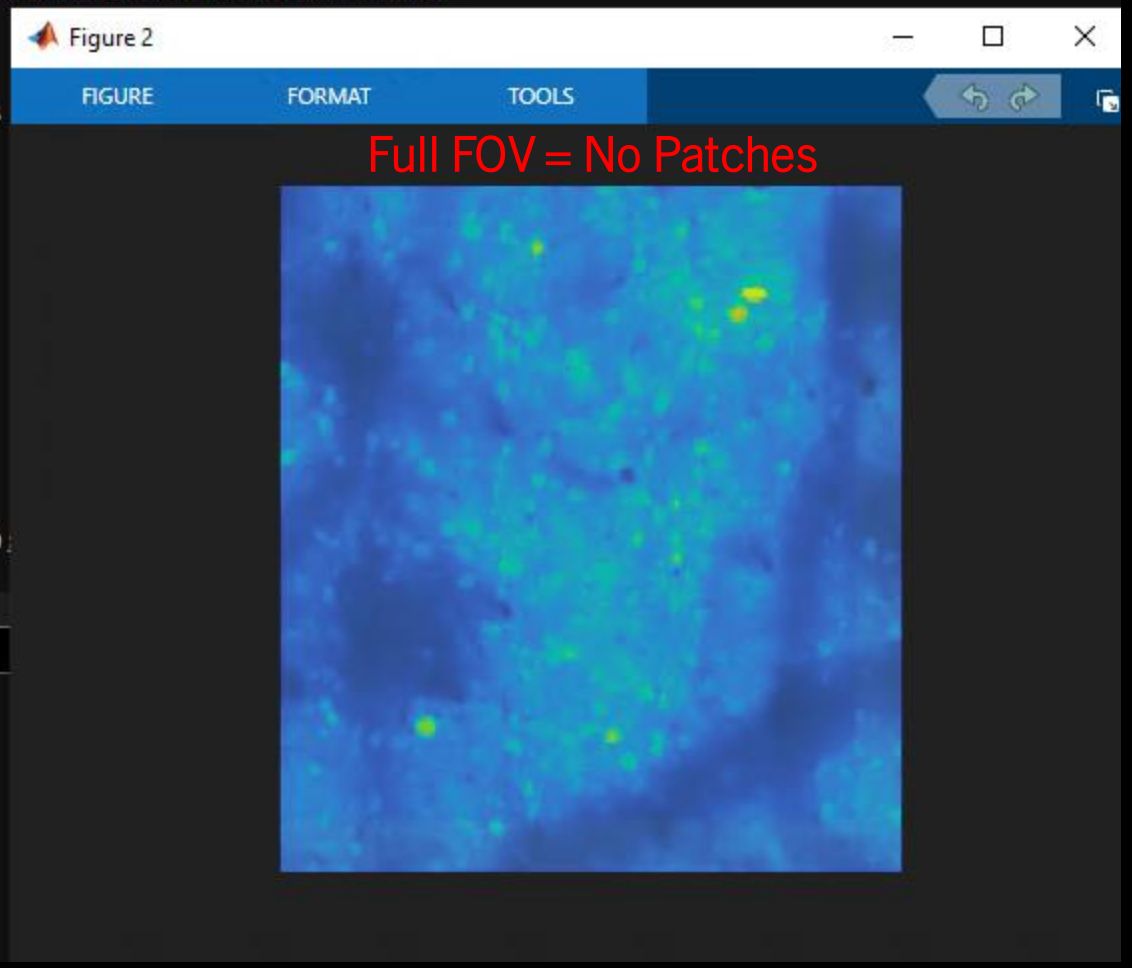
Visualize patch sizes

notebooks/segmentation.m

```
mx = ceil(pi.*(1.33.*tau).^2);
mn = floor(pi.*(tau.*0.5).^2);

% patch set up: basing it on the 600 um strips of the 2pRAM, +50 um overlap between patches
fov = metadata.fov(1);
patch_size = round(fov).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% Visualize Patches

figure;
img = h5read(motion_corrected_filename, '/Ym');

imagesc(img); axis image; axis off;
hold on;

for k = 1:length(patches)
    patch = patches{k};
    x = [patch(3), patch(4), patch(4), patch(3)];
    y = [patch(1), patch(1), patch(2), patch(2)];
    fill(x, y, 'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none');
end
```

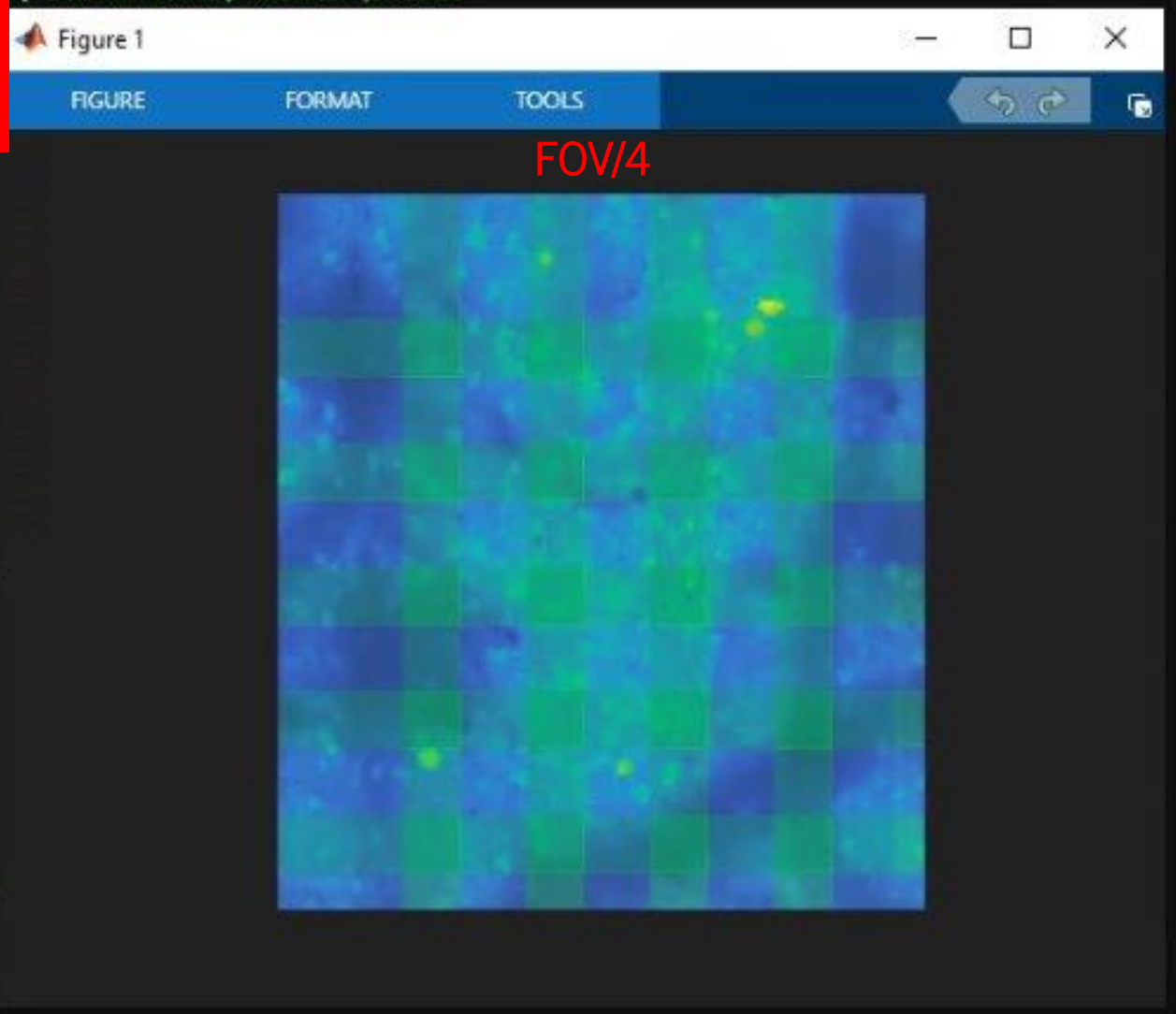
Window

```
Attempt made to read an unknown dataset: /Y.
g with the root dataset '/'...
data found in the root '/' group of the dataset.
Attempt made to read an unknown dataset: /Y.
g with the root dataset '/'...
data found in the root '/' group of the dataset.
Attempt made to read an unknown dataset: /Y.
```



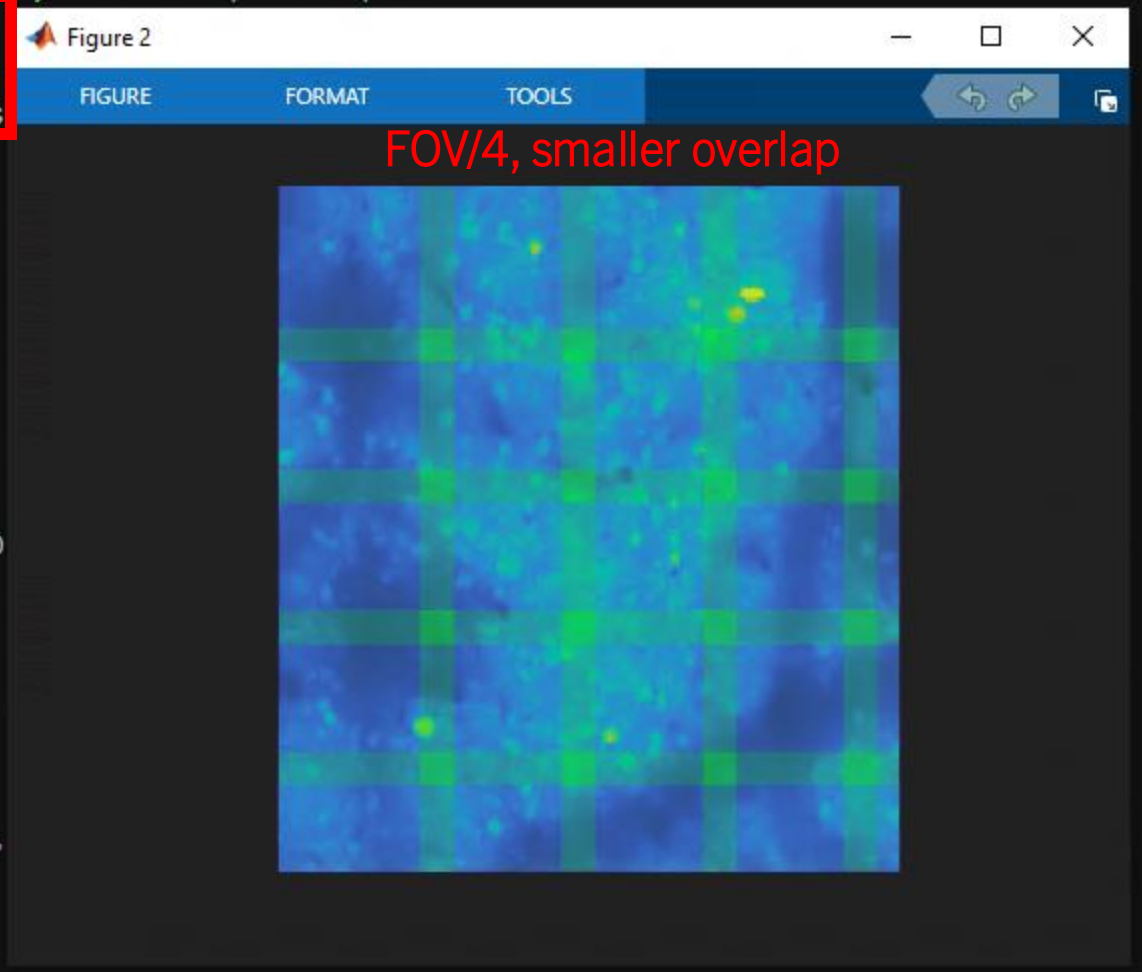
notebooks/segmentation.m

```
% patch set up: basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
fov = metadata.fov(1);
patch_size = round(fov/4).*[1,1];
overlap = [1,1].*ceil(50./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% Visualize Patches

figure;
img = h5read(motion_corrected_filename, '/Ym');

imagesc(img); axis image; axis off;
hold on;

for k = 1:length(patches)
    patch = patches{k};
    x = [patch(3), patch(4), patch(4), patch(3)];
    y = [patch(1), patch(1), patch(2), patch(2)];
    fill(x, y, 'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none')
end

% Draw overlaps
for k = 1:length(patches)
    patch = patches{k};
    for i = patch(1):overlap(1):patch(2)-overlap(1)
        for j = patch(3):overlap(2):patch(4)-overlap(2)
            x = [j, j + overlap(2), j + overlap(2), j];
            y = [i, i, i + overlap(1), i + overlap(1)];
            fill(x, y, 'b', 'FaceAlpha', 0.05, 'EdgeColor'
        end
    end
end
```



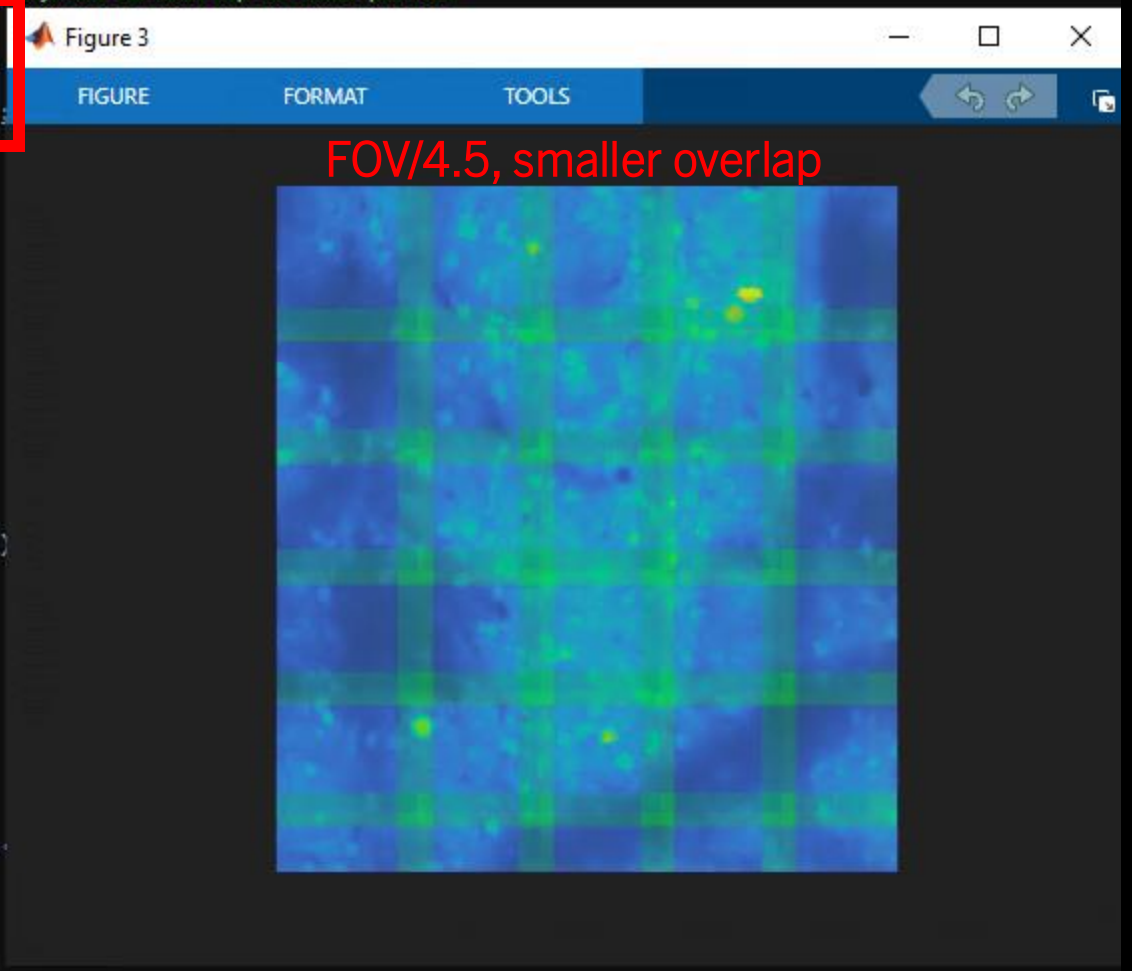
notebooks/segmentation.m

```
% patch set up: basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
fov = metadata.fov(1);
patch_size = round(fov/4).*[1,1];
overlap = [1,1].*ceil(30./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% Visualize Patches

figure;
img = h5read(motion_corrected_filename, '/Ym');

imagesc(img); axis image; axis off;
hold on;

for k = 1:length(patches)
    patch = patches{k};
    x = [patch(3), patch(4), patch(4), patch(3)];
    y = [patch(1), patch(1), patch(2), patch(2)];
    fill(x, y, 'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none')
end

% Draw overlaps
for k = 1:length(patches)
    patch = patches{k};
    for i = patch(1):overlap(1):patch(2)-overlap(1)
        for j = patch(3):overlap(2):patch(4)-overlap(2)
            x = [j, j + overlap(2), j + overlap(2), j];
            y = [i, i, i + overlap(1), i + overlap(1)];
            fill(x, y, 'b', 'FaceAlpha', 0.05, 'EdgeColor'
        end
    end
end
```



notebooks/segmentation.m

```
% patch set up: basing it on the ~600 um strips of the 2pRAM, +50 um overlap between patches
fov = metadata.fov(1);
patch_size = round(fov/4.5).*[1,1];
overlap = [1,1].*ceil(30./pixel_resolution);
patches = construct_patches(data_size,patch_size,overlap);

% Visualize Patches

figure;
img = h5read(motion_corrected_filename, '/Ym');

imagesc(img); axis image; axis off;
hold on;

for k = 1:length(patches)
    patch = patches{k};
    x = [patch(3), patch(4), patch(4), patch(3)];
    y = [patch(1), patch(1), patch(2), patch(2)];
    fill(x, y, 'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none')
end

% Draw overlaps
for k = 1:length(patches)
    patch = patches{k};
    for i = patch(1):overlap(1):patch(2)-overlap(1)
        for j = patch(3):overlap(2):patch(4)-overlap(2)
            x = [j, j + overlap(2), j + overlap(2), j];
            y = [i, i, i + overlap(1), i + overlap(1)];
            fill(x, y, 'b', 'FaceAlpha', 0.05, 'EdgeColor'
        end
    end
end
```



notebooks/segmentation.m

First pass of CNMF

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc

%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;

%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);

%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')

%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);

%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)

%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);

%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                  % fluorescence
Fd = prctfilt(F,df_percentile,window);                 % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);  % background + baseline for each component
F_dff = Fd./Bc;

%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*

notebooks/segmentation.m

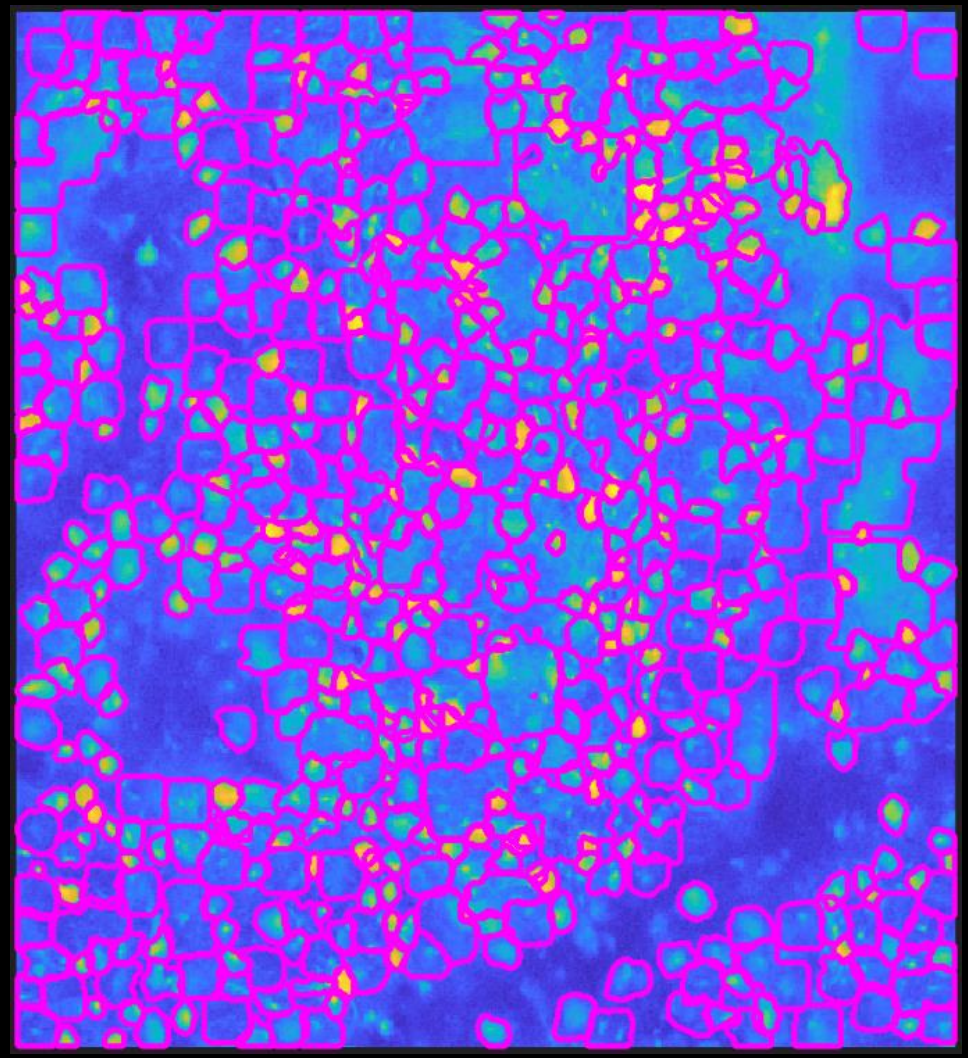
First pass of CNMF

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc

%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;

%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);

%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')

%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);

%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)

%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);

%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                    % fluorescence
Fd = prctfilt(F,df_percentile,window);                   % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);    % background + baseline for each component
F_dff = Fd./Bc;

%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

View masks of first pass

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*



notebooks/segmentation.m

First pass of CNMF

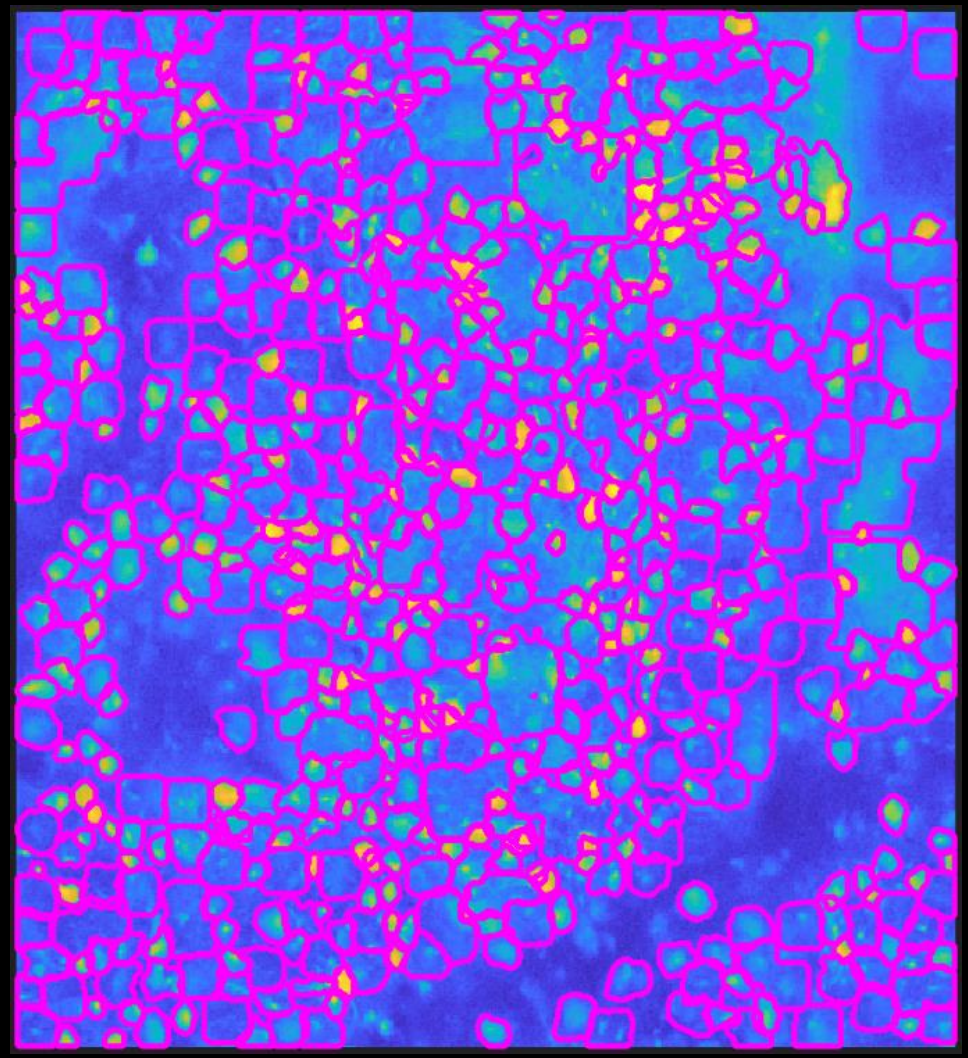
View masks of first pass

Refine selection

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc

%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;

%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);

%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')

%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);

%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)

%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);

%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                    % fluorescence
Fd = prctfilt(F,df_percentile,window);                   % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);    % background + baseline for each component
F_dff = Fd./Bc;

%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*



notebooks/segmentation.m

First pass of CNMF

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc
```

View masks of first pass

```
%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;
```

Refine selection

```
%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);
```

View selected / rejected masks

```
%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')
```

```
%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);
```

```
%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)
```

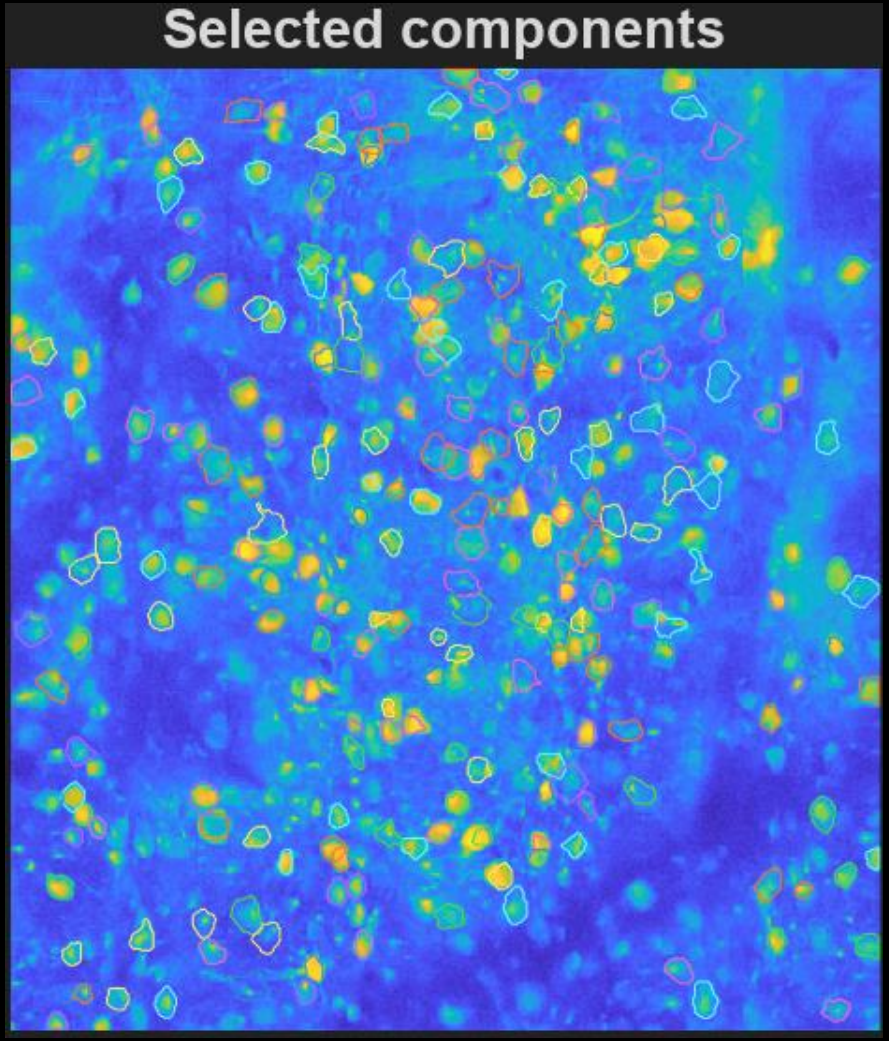
```
%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);
```

```
%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                    % fluorescence
Fd = prctfilt(F,df_percentile,window);                   % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);    % background + baseline for each component
F_dff = Fd./Bc;
```

```
%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*



notebooks/segmentation.m

First pass of CNMF

View masks of first pass

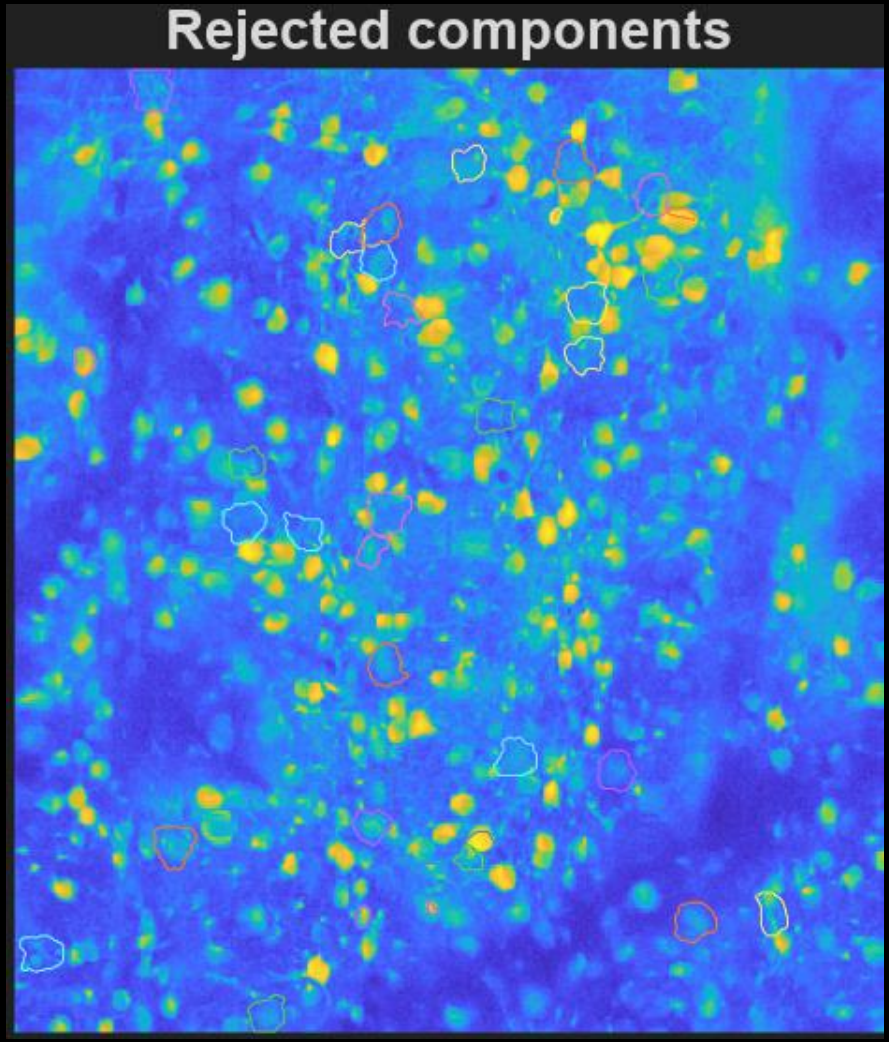
Refine selection

View selected / rejected masks

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc

%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;

%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);

%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')

%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);

%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)

%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);

%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                    % fluorescence
Fd = prctfilt(F,df_percentile,window);                   % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);    % background + baseline for each component
F_dff = Fd./Bc;

%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*



notebooks/segmentation.m

First pass of CNMF

View masks of first pass

Refine selection

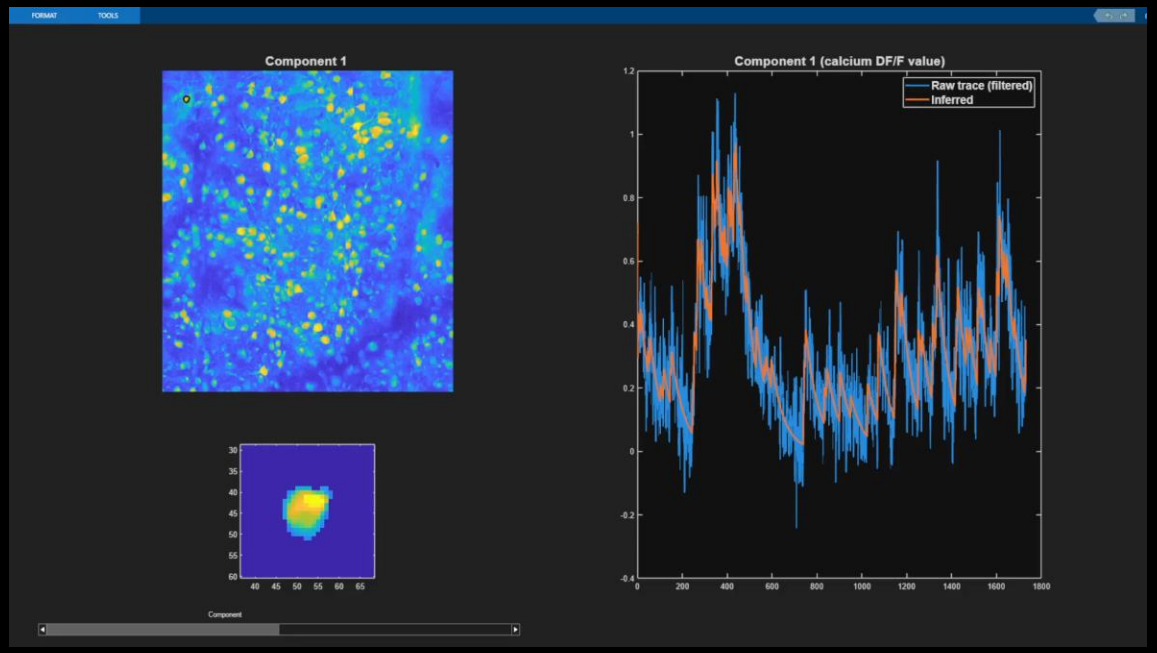
View selected / rejected masks

Use selected neurons to update traces

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc

%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;

%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);

%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')

%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);

%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)

%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);

%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                    % fluorescence
Fd = prctfilt(F,df_percentile,window);                   % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);    % background + baseline for each component
F_dff = Fd./Bc;

%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*



notebooks/segmentation.m

First pass of CNMF

```
%% run CNMF algorithm on patches and combine
tic;
[A,b,C,f,S,P,RESULTS,YrA] = run_CNMF_patches(h5read(motion_corrected_filename, '/Y'),K,patches,tau,p,options);
[ROIvars.rval_space,ROIvars.rval_time,ROIvars.max_pr,ROIvars.sizeA,keep] = classify_components(M_nr,A,C,b,f,YrA,options);
toc
```

View masks of first pass

```
%% a simple GUI

Coor = plot_contours(A,Cn,options,1); close;
```

Refine selection

```
%% Classify / validate components

[rval_space,~,~,sizeA,~,~,traces] = classify_components_jeff(M_nr,A,C,b,f,YrA,options);
ind_corr = (rval_space > space_thresh) & (sizeA >= options.min_size_thr) & (sizeA <= options.max_size_thr);

% Event exceptionality:
fitness = compute_event_exceptionality(traces,options.N_samples_exc,options.robust_std);
ind_exc = (fitness < options.min_fitness);

% Select components:
keep = ind_corr & ind_exc;

A_keep = A(:,keep);
C_keep = C(keep,:);
Km = size(C_keep,1); % total number of components

rVals = rval_space(keep);
```

View selected / rejected masks

```
%% view contour plots of selected and rejected components

throw = ~keep;
figure;
ax1 = subplot(121); plot_contours(A(:,keep),Cn,options,0,[],Coor,1,find(keep)); title('Selected components','fontweight','bold','fontsize',14);
ax2 = subplot(122); plot_contours(A(:,throw),Cn,options,0,[],Coor,1,find(throw));title('Rejected components','fontweight','bold','fontsize',14);
linkaxes([ax1,ax2],'xy')

%% inspect components
plot_components_GUI(M_nr,A(:,keep),C(keep,:),b,f,Cn,options);
plot_components_GUI(Yr,A_or,C_or,b2,f2,Cn,options);

%% make movie

make_patch_video(A(:,keep),C(keep,:),b,f,M_nr,Coor,options)
```

Use selected neurons to update traces

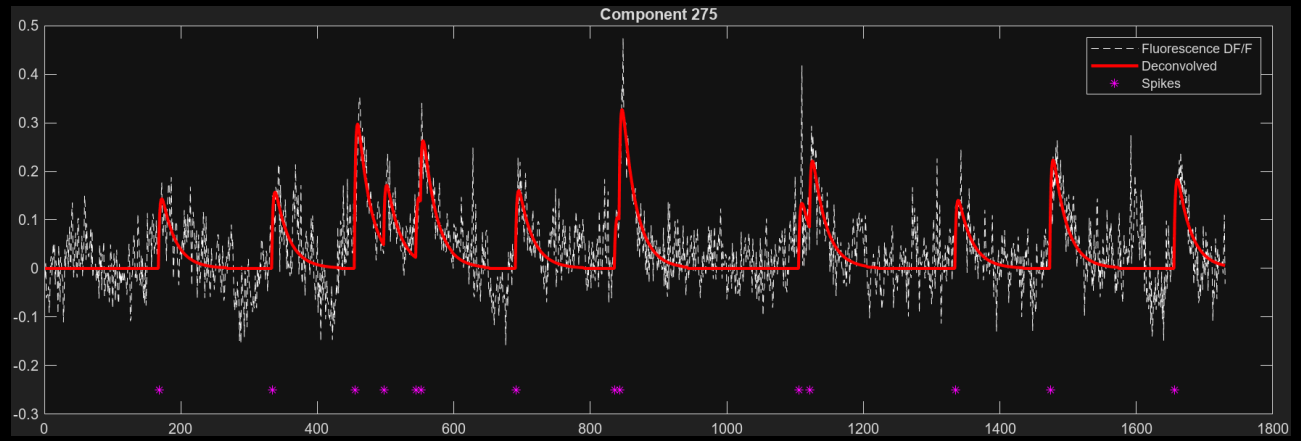
```
%% refine temporal components
A_keep = A(:,keep);
C_keep = C(keep,:);
[C2,f2,P2,S2,YrA2] = update_temporal_components(reshape(M_nr,[],T),A_keep,b,C_keep,f,P,options);

%% detrend fluorescence and extract DF/F values

df_percentile = 30;
window = 1000;

F = diag(sum(A_keep.^2))*(C2 + YrA2);                  % fluorescence
Fd = prctfilt(F,df_percentile,window);                 % detrended fluorescence
Bc = prctfilt((A_keep'*b)*f2,30,1000,300,0) + (F-Fd);  % background + baseline for each component
F_dff = Fd./Bc;
```

Evaluate spike estimates

```
%% deconvolve data

nNeurons = size(F_dff,1);
C_dec = zeros(size(F_dff));
S = zeros(size(F_dff));
kernels = cell(nNeurons,1);
min_sp = 3;    % find spikes resulting in transients above min_sp x noise level
```

Remainder of notebook

- Same workflow as segmentPlane()
- Runs a single z-plane
- *Non-sparse representation (interactive)*



Segmentation outputs:

`T_keep` : neuronal time series [Km, T]. `single`

`Ac_keep` : neuronal footprints [2*tau+1, 2*tau+1, Km]. `single`

`C_keep` : denoised time series [Km, T]. `single`

`Km` : number of neurons found. `single`

`Cn` : correlation image [x, y]. `single`

`b` : background spatial components [x*y, 3]. `single`

`f` : background temporal components [3, T]. `single`

`acx` : centroid in x direction for each neuron [1, Km]. `single`

`acy` : centroid in y direction for each neuron [1, Km]. `single`

`acm` : sum of component pixels for each neuron [1, Km]. `single`

LBM-CaImAn

PIPELINE STEPS

1 Assembly | 2 Motion Correction | 3 Segmentation | 4 **Collation**

data_path

- Points to save_path used in segmentPlane()
  - Checks for the presence of segmented_plane_Z_i
- Output filename will always be caiman_output_minSNR_X_plane_z_i.mat

planez_i for i =start_plane:end_plane
X = options.min_SNR

4. collatePlanes()

LBM-CaImAn

PIPELINE STEPS

1 Assembly

2 Motion Correction

3 Segmentation

4 **Collation**

Before proceeding

You will need access to the following calibration files:

`pollen_calibration_Z_vs_N.mat` `pollen_calibration_x_y_offsets.fig`

Important

These files hold data used to align each z-plane depth around the same [Y, X] coordinates. Place these files in the same directory as your `segmentation_plane_N` files.

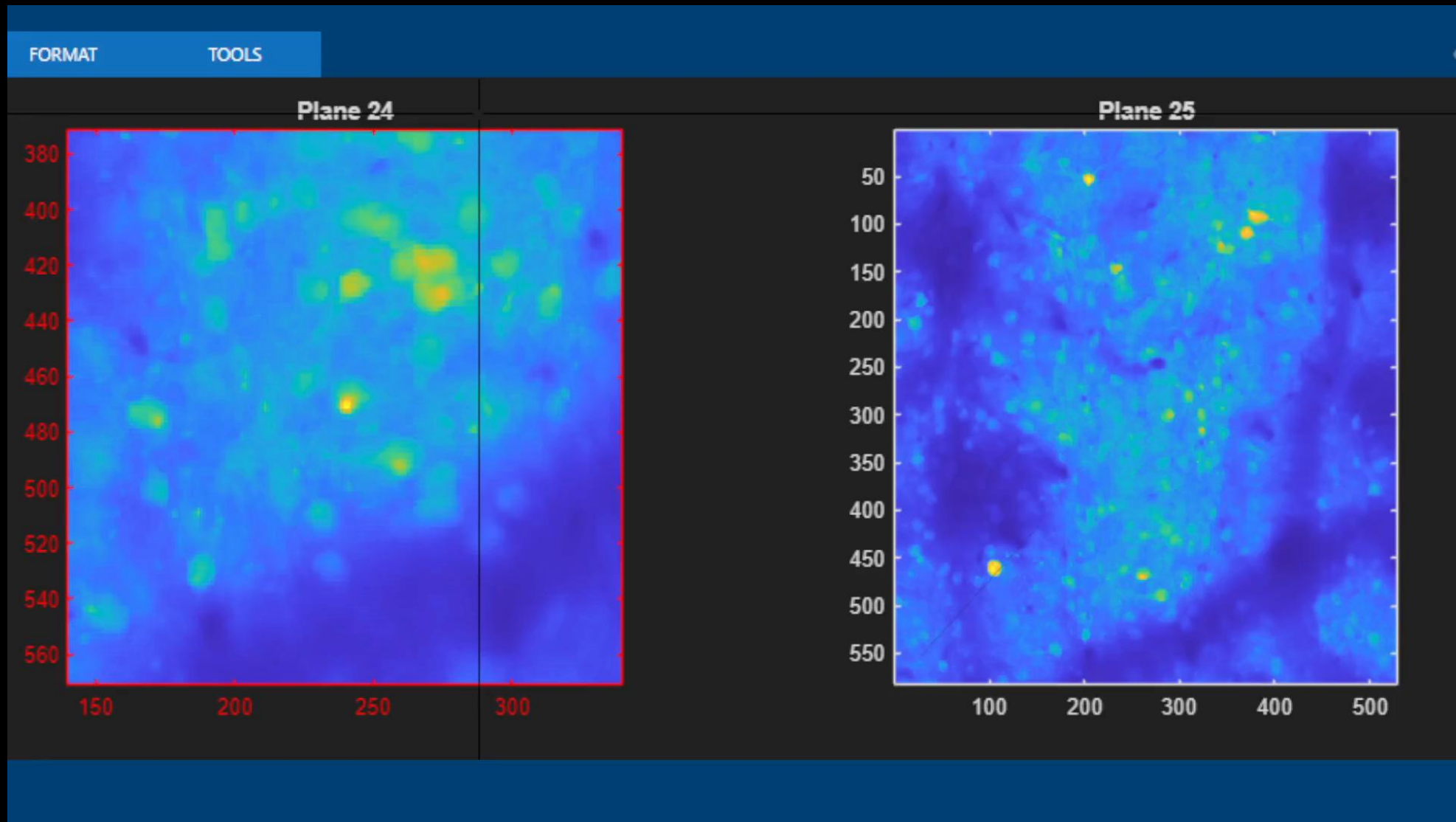
4. `collatePlanes()`

_plane_z$_i$ for i =start_plane:end_plane
X = options.min_SNR

Function-Specific Parameters:

- **num_cores** (`double, integer, positive`) – Number of cores to use for computation. The value is limited to a maximum of 24 cores.
- **start_plane** (`double, integer, positive`) – The starting plane index for processing.
- **end_plane** (`double, integer, positive`) – The ending plane index for processing. Must be greater than or equal to start_plane.
- **num_features** (`double, positive`) – The number of neurons to select for the lateral offset correction between z-planes. Default is 1 feature/neuron.
- **options** (`struct`) – Struct containing key-value pairs for ‘min_SNR’ and ‘merge_thresh’.
- **motion_corrected_path** (`string`) – Path to motion corrected data. Default is data_path/../motion_corrected/,

demo_LBM_pipeline.m

```
%% 4) Collate z-planes and merge spatially overlapping highly-correlated neuronss
thresh_options = {'min_SNR', 1.5, 'merge_thr', 0.95};
collatePlanes( ...
    'C:/Users/RBO/caiman_data/animal_01/session_01/segmented', ... % data_path = segmentation results
    'ds', '/Y', ...
    'debug_flag', 0, ...
    'overwrite', 0, ...
    'start_plane', 1, ...
    'end_plane', metadata.num_planes,  ...
    'num_features', 1, ...
    'motion_correction_path', fullfile(data_path, 'motion_corrected/'), ...
    'options', thresh_options ...
);
```

Correct lateral offsets between adjacent z-planes