

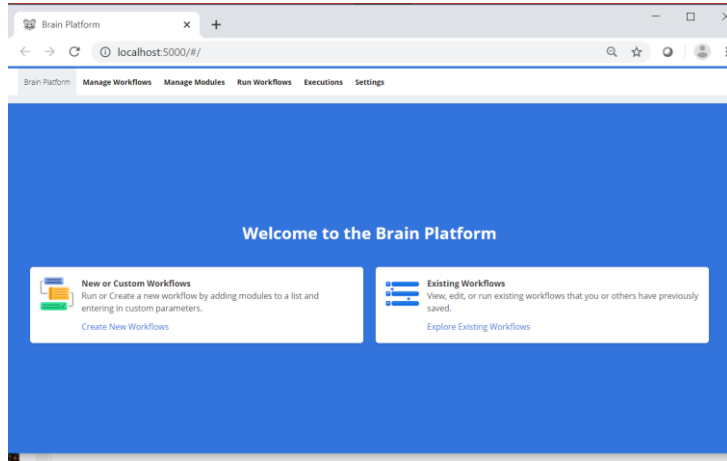
# BRAIN Platform

FRAUNHOFER CESE  
UMD

# BRAIN Platform Overview

## Manage, Share, and Validate Workflow

- Web-based GUI
- Interface with cloud storage (Amazon AWS) to pull contents
- HDF5-based file system (enable NWB)



## Key features:

- Streamlines the process of collaborating and sharing data, algorithms and analysis pipelines
- Adaptive workflow management: Integrates multiple open source and custom built calcium imaging tools in multiple languages

# BRAIN Platform - Creating Workflow

Modules currently integrated within the BRAIN platform – A workflow can be setup using **any** combination of modules.

Brain Platform [Manage Workflows](#) [Manage Modules](#) [Run Workflows](#) [Executions](#) [Settings](#)

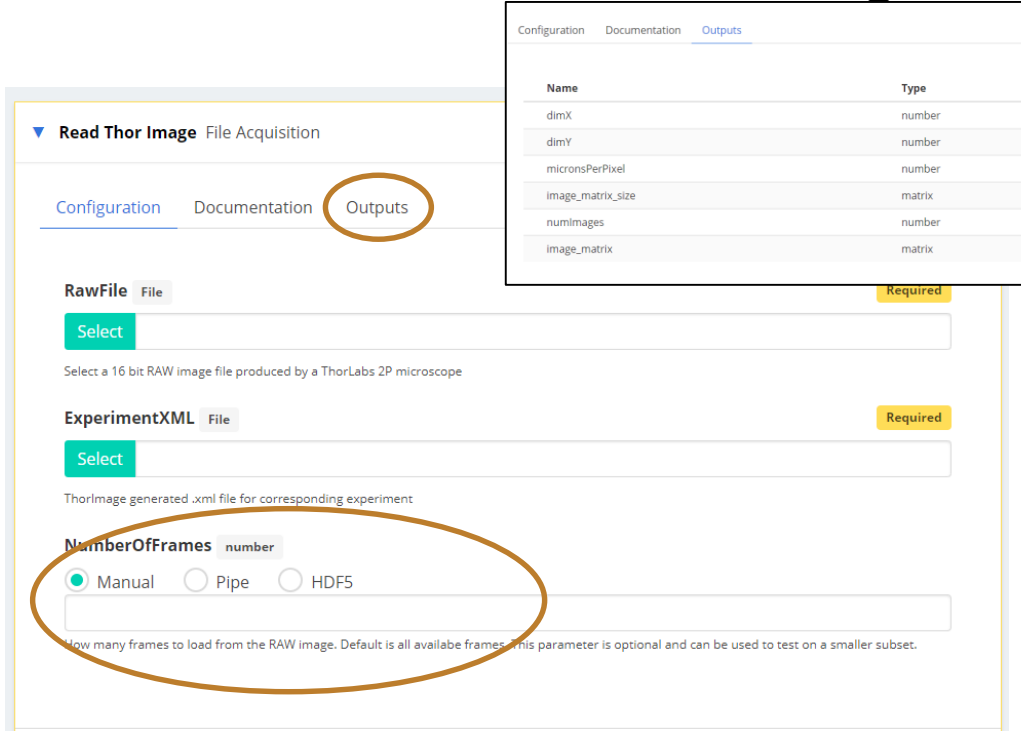
## Create Workflow



[All](#) [File Acquisition](#) [Motion Correction](#) [Cell Selection](#) [Signal Detection](#) [Analysis](#)

Name	Type	Language	Description	
Automatic Cell Registration	Cell Selection	matlab		<a href="#">Add</a>
Compute DFF	Signal Detection	matlab		<a href="#">Add</a>
Single-step DFT Image Registration	Motion Correction	matlab		<a href="#">Add</a>
Manual Cell Selection	Cell Selection	matlab		<a href="#">Add</a>
Motion Correction	Motion Correction	matlab		<a href="#">Add</a>
Read NWB	File Acquisition	matlab	Read NWB data from files	<a href="#">Add</a>
Read Prairie	File Acquisition	matlab	Read Prairie image files	<a href="#">Add</a>
Read Thor Image	File Acquisition	matlab	Read Thor image files	<a href="#">Add</a>
s2pSpikeDeconv	Analysis	python	suite2p spike deconvolution	<a href="#">Add</a>

# BRAIN Platform - Creating Workflow



Configuration Documentation **Outputs**

**RawFile** File Required

Select

Select a 16 bit RAW image file produced by a ThorLabs 2P microscope

**ExperimentXML** File Required

Select

ThorImage generated .xml file for corresponding experiment

**NumberOfFrames** number

Manual  Pipe  HDF5

How many frames to load from the RAW image. Default is all available frames. This parameter is optional and can be used to test on a smaller subset.

Name	Type
dimX	number
dimY	number
micronsPerPixel	number
image_matrix_size	matrix
numImages	number
image_matrix	matrix

Three ways of setting value for the module:

- Manual
- Pipe: Use output of module executed earlier in the pipeline
- HDF5: Use output from an HDF5 (e.g., result of prior executions)







# BRAIN Platform - Executing Workflow

Brain Platform **Manage Workflows** Manage Modules Run Workflows Executions Settings

## Workflow Library

New Workflow



Name ▲	Date Modified	Date Created	
Default Workflow	2020-01-01 12:34 PM	2020-01-01 12:34 PM	  
DFF Analysis	2020-04-14 08:20 PM	2020-04-14 08:20 PM	  

- All created workflows (chain of modules and their configuration values) are listed here.
- **BRAIN platform installed with a default workflow.**
- Currently newly created workflows are stored locally, but capability exists to store and share workflows via cloud storage.

# BRAIN Platform - Executing Workflow

Brain Platform **Manage Workflows** Manage Modules Run Workflows Executions Settings

## Run a Workflow

**Workflows**  
Search  
Default Workflow  
View All Workflows

### Default Workflow

**Run Workflow**

**Experiment Name**  
Default Workflow\_2020-04-14\_20:18:51  
Give a short, descriptive name to the experiment

**Result Directory**  
Default  
Choose where to store the experiment results. You can manage these options in settings

**Read Prairie** File Acquisition

Parameters	Description
<b>TiffDirectory</b> Directory <span>Required</span>	
Select	Path to a directory containing a sequence of Tiff files
<b>XMLFile</b> File <span>Required</span>	

Once a workflow has been set up, it can be reused to run different experiments by simply changing the input data.

Different result directories (e.g., for different type of experiments) can be set up.

# BRAIN Platform - Executions Status

Tracks workflow executions:

- Ongoing
- Pending (execution queue)
- Completed: Successful & Fail

Categorizes executions based on where results are stored

**Execution History**  
Monitor experiment executions and view results

### Active Executions

Name	Workflow	Execution ID	Status
My DFF Analysis_test3	My DFF Analysis	5c7e27f2-de83-40d1-b618-7b3473df9526	
My DFF Analysis_test4	My DFF Analysis	bbf1fd3b-c77b-40f8-be69-3fc44e2b30c2	waiting

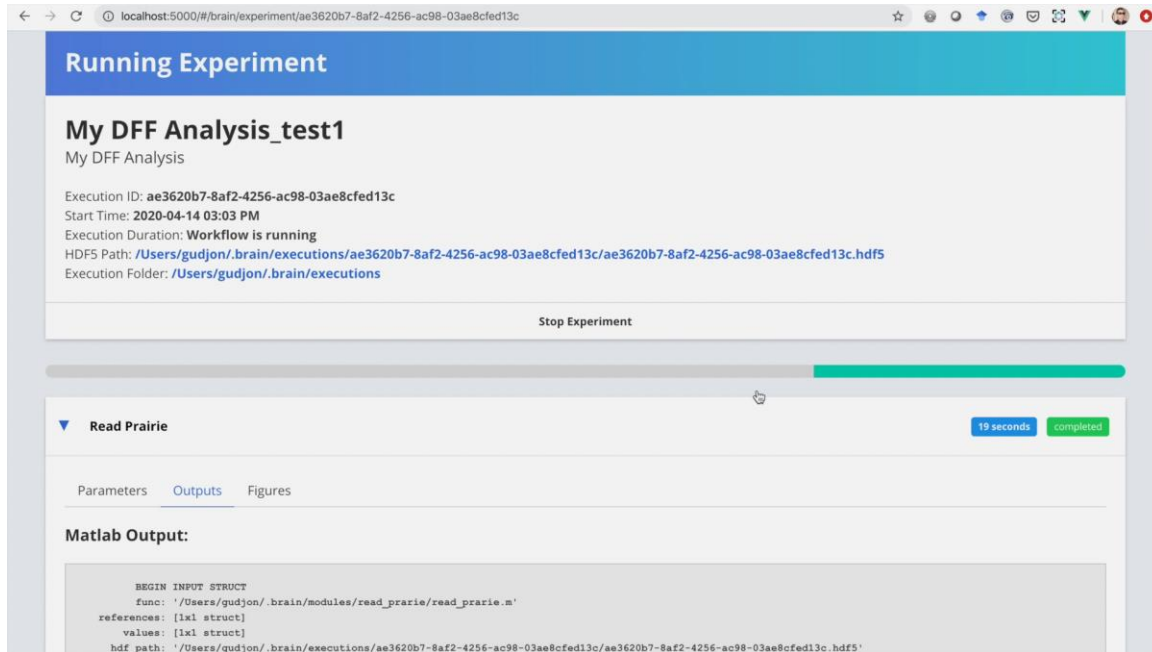
### Execution History

All **Default** Local Test Lab Network Drive

Name	Workflow	Date ▼	Execution ID	Status
My DFF Analysis_test2	My DFF Analysis	2020-04-14 03:07 PM	6bd507bc-60ef-4ca8-8609-a84d401d4779	completed
My DFF Analysis_test1	My DFF Analysis	2020-04-14 03:03 PM	ae3620b7-8af2-4256-ac98-03ae8cfed13c	completed
My DFF Analysis_2020-04-	My DFF	2020-04-14	21c55994-a8fd-4f3d-918d-	failed

# BRAIN Platform - Executions Status

Shows relevant information about execution: experiment configuration, status, location of result.



The screenshot shows a web browser window displaying the execution status page for an experiment. The page has a blue header with the text "Running Experiment". Below the header, the experiment name "My DFF Analysis\_test1" is displayed, followed by "My DFF Analysis". The execution details include: Execution ID: ae3620b7-8af2-4256-ac98-03ae8cfed13c, Start Time: 2020-04-14 03:03 PM, Execution Duration: Workflow is running, HDF5 Path: /Users/gudjon/.brain/executions/ae3620b7-8af2-4256-ac98-03ae8cfed13c/ae3620b7-8af2-4256-ac98-03ae8cfed13c.hdf5, and Execution Folder: /Users/gudjon/.brain/executions. A "Stop Experiment" button is visible. Below this, a progress bar shows the execution is 19 seconds completed. The "Read Prairie" module is expanded, showing its status as "19 seconds completed". The "Matlab Output" section displays the following code:

```
BEGIN INPUT STRUCT
func: '/Users/gudjon/.brain/modules/read_prairie/read_prairie.m'
references: [1x1 struct]
values: [1x1 struct]
hdf_path: '/Users/gudjon/.brain/executions/ae3620b7-8af2-4256-ac98-03ae8cfed13c/ae3620b7-8af2-4256-ac98-03ae8cfed13c.hdf5'
```

The video shows an example of an execution status page.

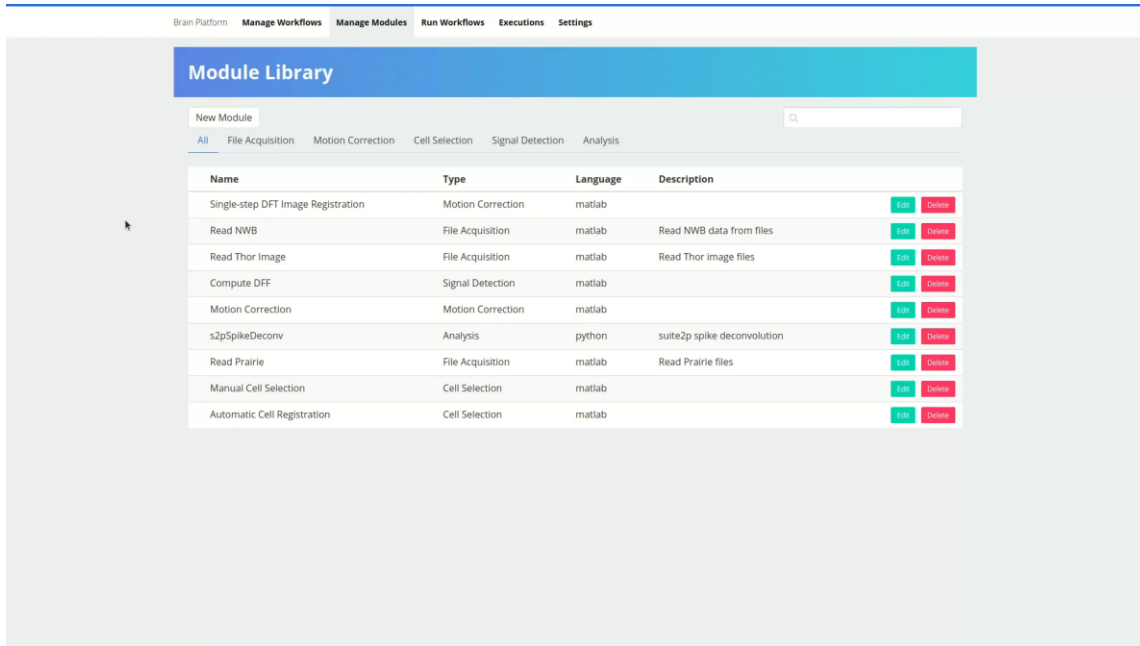
The progress of each module's status is shown (in progress, waiting, completed and time elapsed).

Figures that are generated by the modules can also be viewed in this page.



# BRAIN Platform - Manage Modules

- Enables addition of new modules or editing existing modules. Addition or change only applied locally.
- Useful to add “small” script to bridge the gap between existing modules, e.g., transposing matrix.



The screenshot displays the 'Manage Modules' section of the BRAIN Platform. At the top, there is a navigation bar with tabs for 'Brain Platform', 'Manage Workflows', 'Manage Modules', 'Run Workflows', 'Executions', and 'Settings'. Below this is a 'Module Library' header with a 'New Module' button and a search bar. A list of modules is shown, each with a name, type, language, description, and edit/delete buttons.

Name	Type	Language	Description		
Single-step DFT Image Registration	Motion Correction	matlab		Edit	Delete
Read NWB	File Acquisition	matlab	Read NWB data from files	Edit	Delete
Read Thor Image	File Acquisition	matlab	Read Thor image files	Edit	Delete
Compute DFF	Signal Detection	matlab		Edit	Delete
Motion Correction	Motion Correction	matlab		Edit	Delete
s2pSpikeDeconv	Analysis	python	suite2p spike deconvolution	Edit	Delete
Read Prairie	File Acquisition	matlab	Read Prairie files	Edit	Delete
Manual Cell Selection	Cell Selection	matlab		Edit	Delete
Automatic Cell Registration	Cell Selection	matlab		Edit	Delete

The video shows the step of creating a simple module with two inputs and one output.

The code template is automatically generated given the inputs and output specification.

# Team Members (Fraunhofer CESE)



Gudjon Magnusson  
[gmagnusson@fc-md.umd.edu](mailto:gmagnusson@fc-md.umd.edu)  
Fraunhofer CESE



Madeline Diep  
[mdiep@fc-md.umd.edu](mailto:mdiep@fc-md.umd.edu)  
Fraunhofer CESE