PLOTTING ALL DATASETS OF CALCIUM ACTIVITY IN THE SMH MUTANT. ACTIVITY DURING THE ENTIRE RECORDING

Experiments done by D'Gama et al, 2photon in HuC:GCamp6s, 5 dpf in smh control (heterozygous and wt) and smh homozygous. Geneotype was based on the curved body axis.

Cells were segmented and DFF recovered. DFF was calculated by removing the average of the entire trace.

Cells were clustered using Pearson's correlations

Traces and Cell segmentations were saved in the datasets below

```
folderPath='X:\Manuscripts\Dgama etal 2024 smh\data\Supplementary Fig 2p\clustered data\'
```

```
folderPath =
'X:\Manuscripts\Dgama etal 2024 smh\data\Supplementary Fig 2p\clustered data\'
```

Load datasets and plot a heatmap

This code will upload the dataset and plot heatmap that are represented in Figure Supplemental 2p.

The results matrix contains

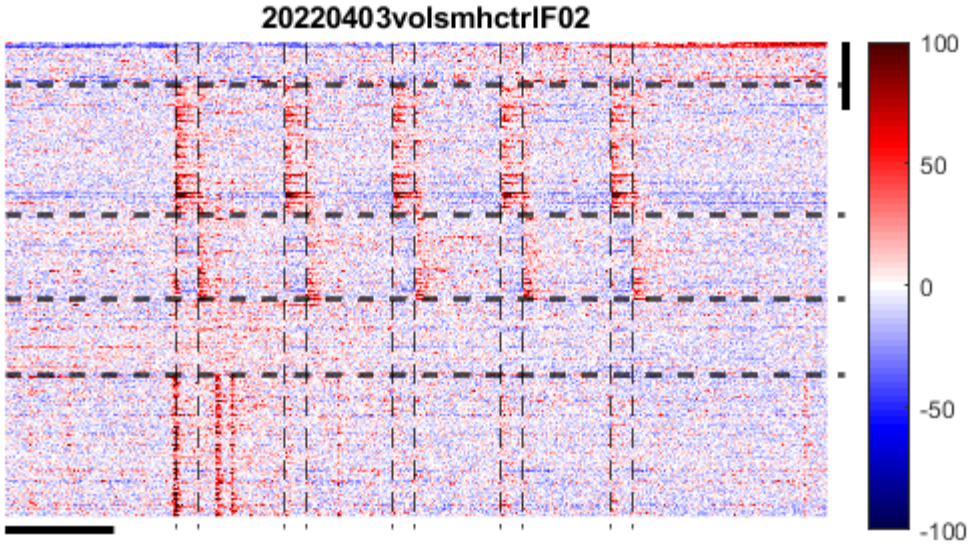
1. the raw trace: DV_DFF_XYZINDEXPLANE_rawtraces
2. the DFF: DV_DFFaveragetime
3. the parameters used for the pearson correlation in cfg
4. the segmented nuclei in neuronLabels
5. the brain region index of each segmented cell: redoneIndex
6. teh metadata of the experiment: metadata

The code below will plot the DFF traces for all control datasets

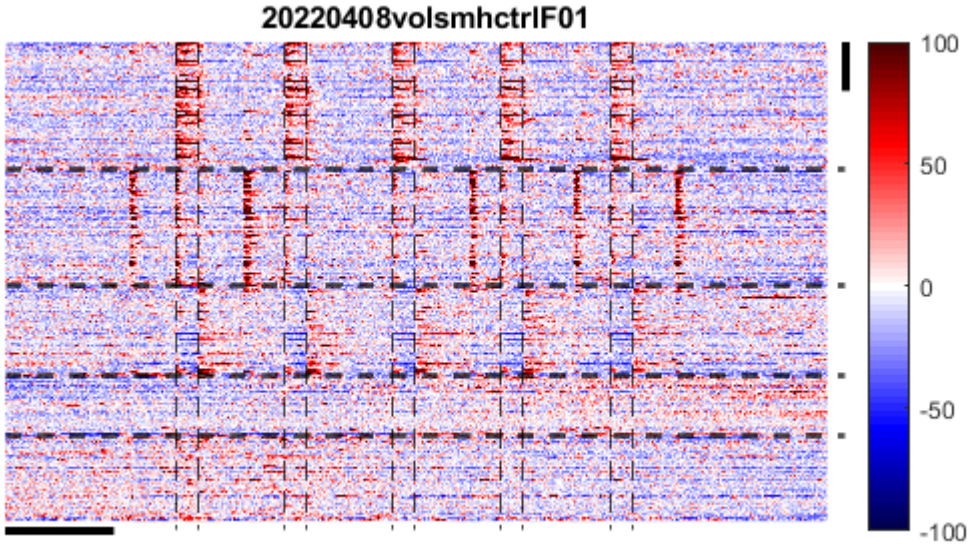
```
% Get the list of files and folders in the specified folder
fileList = dir(folderPath);

% Loop through each file in the folder
for i = 1:length(fileList)
    % Skip directories
    if ~fileList(i).isdir
        if contains(fileList(i).name, 'ctrl')
        % Get the full path of the file
        filePath = fullfile(folderPath, fileList(i).name);

        % Display the file name
        disp(['Opening file: ', fileList(i).name]);

        % Open the file and read its contents
        % Assuming the files are text files, you can use fopen, fread, etc.
        fileID = fopen(filePath, 'r');
        if fileID == -1
            disp(['Failed to open file: ', fileList(i).name]);
        else
```

```
            % Read the file content (assuming text file here)
            load(filePath);
            figure, plot_heatmap(results, folderPath)

            % Add your file processing code here
        end
    end
    end
end
```

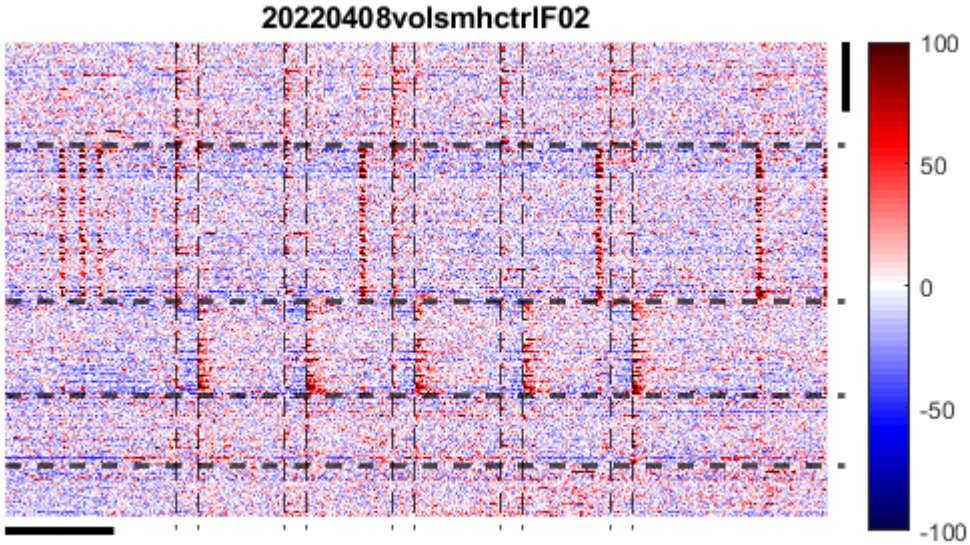
```
Opening file: 20220403volsmhctrlF02Results_CLUSTERING.mat
```



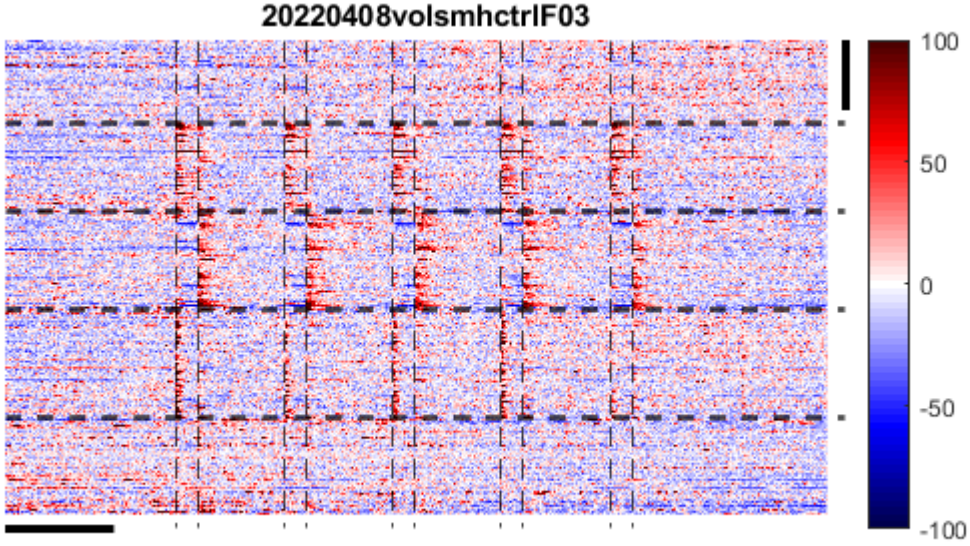

```
Opening file: 20220408volsmhctrlF01Results_CLUSTERING.mat
```



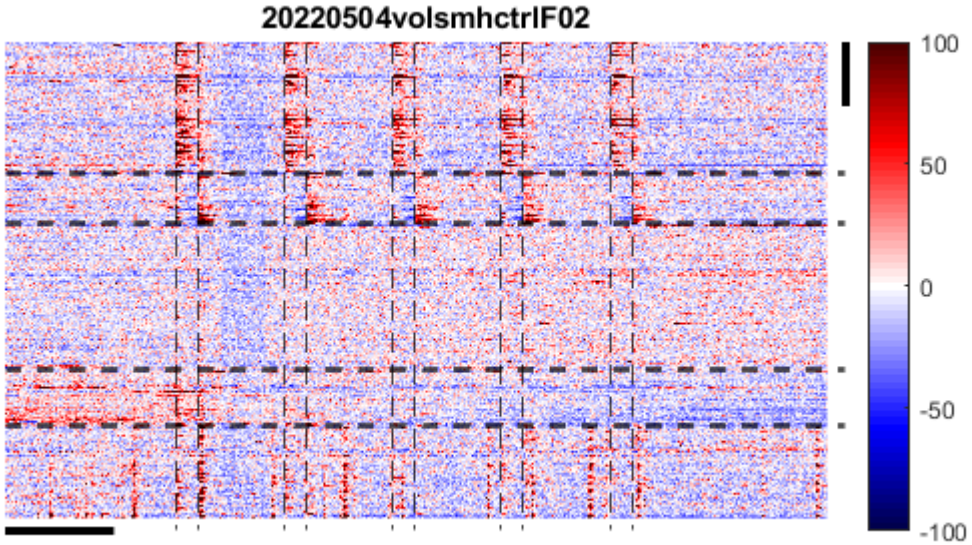

```
Opening file: 20220408volsmhctrlF02Results_CLUSTERING.mat
```



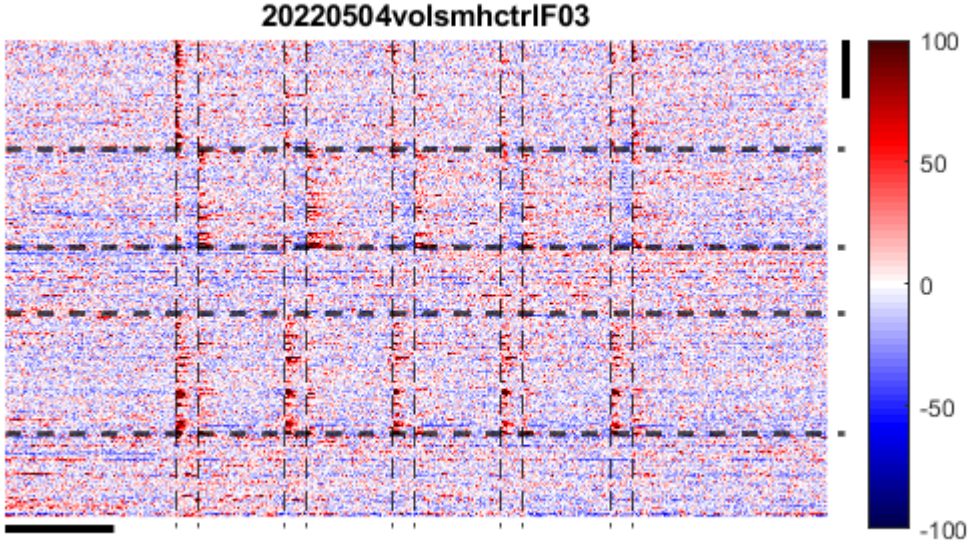
Opening file: 20220408volsmhctrlF03Results_CLUSTERING.mat



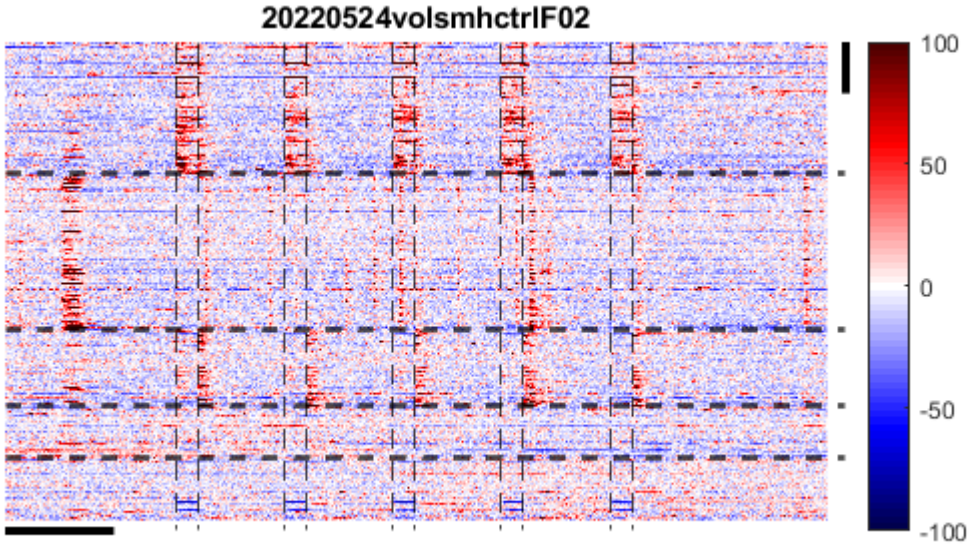
Opening file: 20220504volsmhctrlF02Results_CLUSTERING.mat



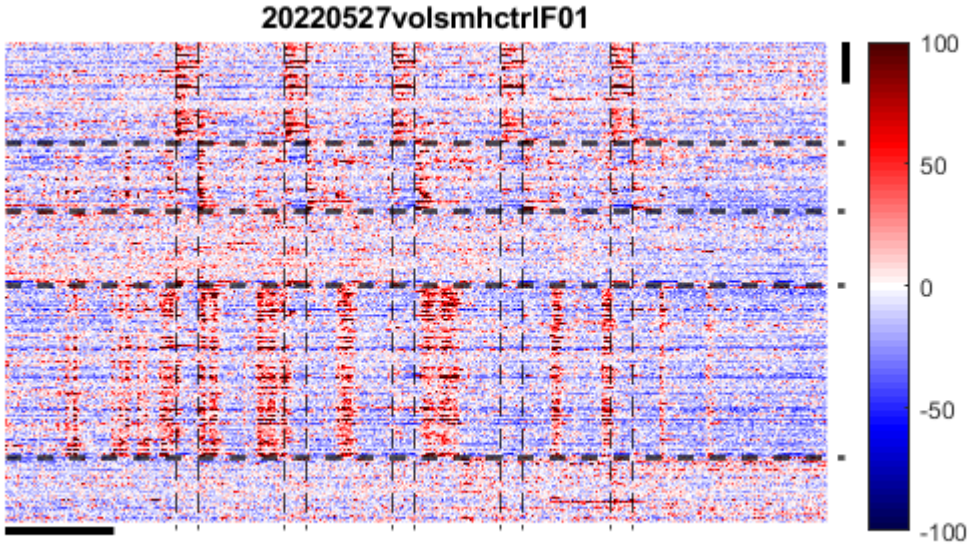
Opening file: 20220504volsmhctrlF03Results_CLUSTERING.mat



Opening file: 20220524volsmhctrlF02Results_CLUSTERING.mat




```
Opening file: 20220527volsmhctrlF01Results_CLUSTERING.mat
```

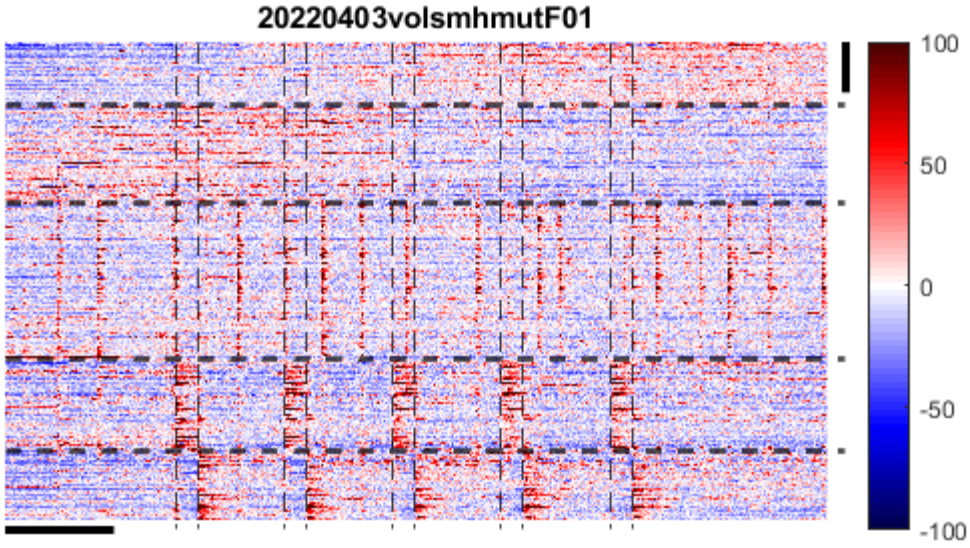


For all mutant datasets

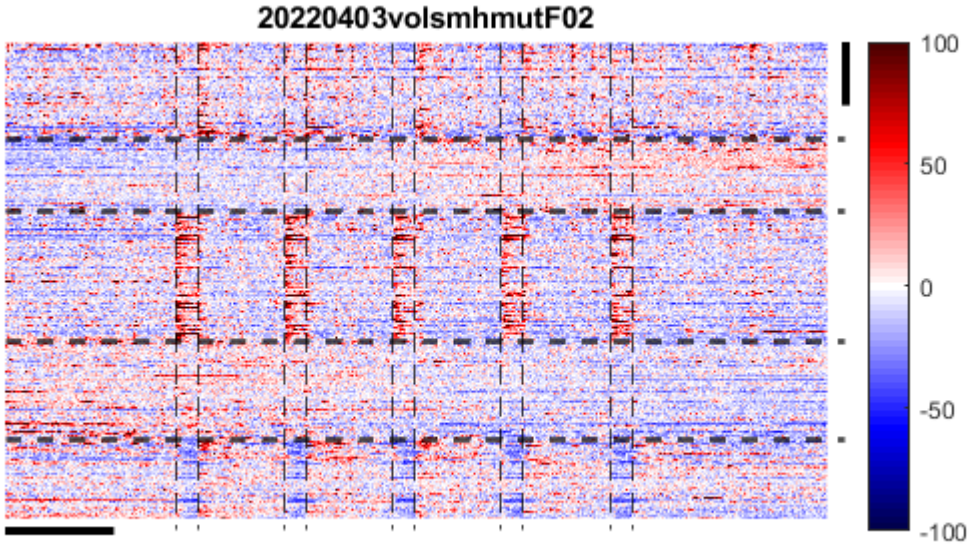
```
% Get the list of files and folders in the specified folder
fileList = dir(folderPath);

% Loop through each file in the folder
for i = 1:length(fileList)
    % Skip directories
    if ~fileList(i).isdir
        if contains(fileList(i).name, 'mut')
        % Get the full path of the file
        filePath = fullfile(folderPath, fileList(i).name);
```

```
        % Display the file name
        disp(['Opening file: ', fileList(i).name]);

        % Open the file and read its contents
        % Assuming the files are text files, you can use fopen, fread, etc.
        fileID = fopen(filePath, 'r');
        if fileID == -1
            disp(['Failed to open file: ', fileList(i).name]);
        else
            % Read the file content (assuming text file here)
            load(filePath);
            figure, plot_heatmap(results, folderPath)

            % Add your file processing code here
        end
    end
    end
end
```

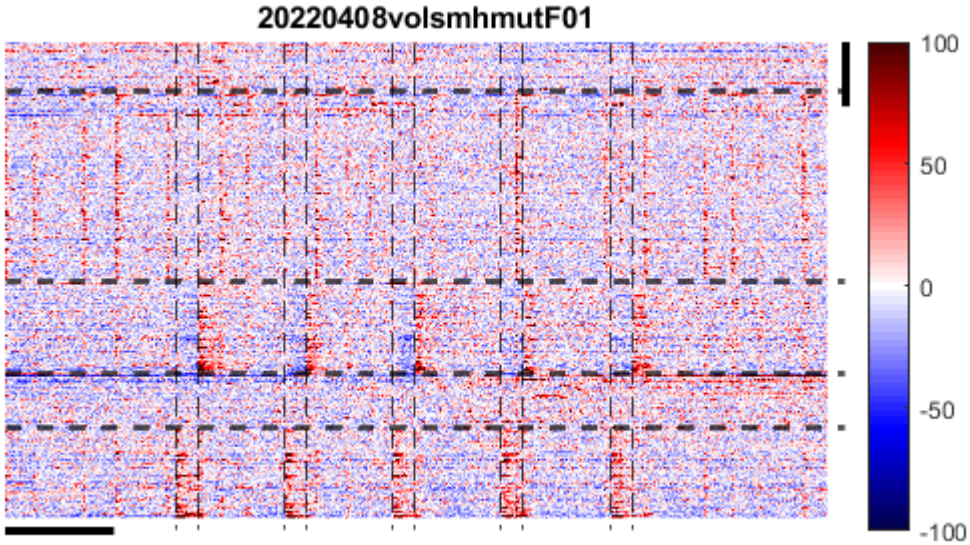
```
Opening file: 20220403volsmhmutF01Results_CLUSTERING.mat
```



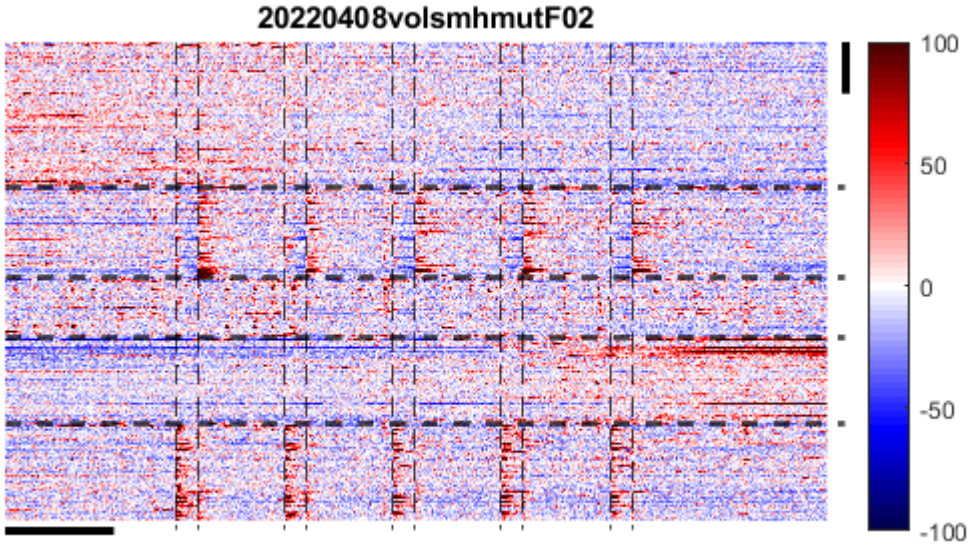

```
Opening file: 20220403volsmhmutF02Results_CLUSTERING.mat
```



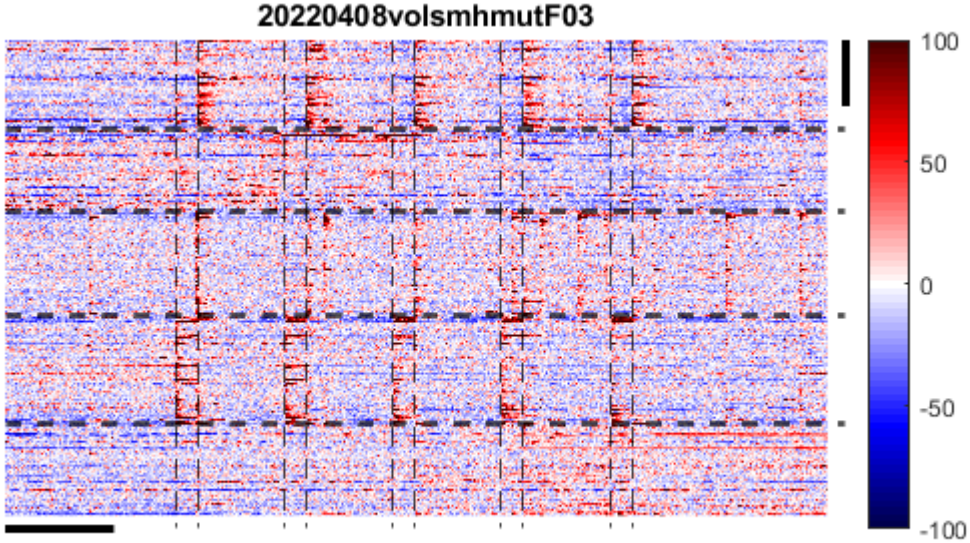
Opening file: 20220408volsmhmutF01Results_CLUSTERING.mat



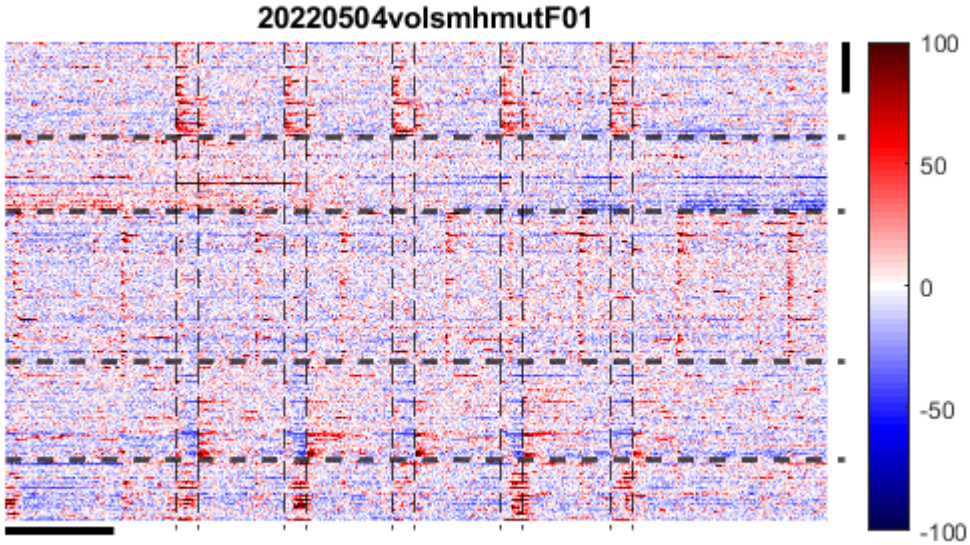
Opening file: 20220408volsmhmutF02Results_CLUSTERING.mat



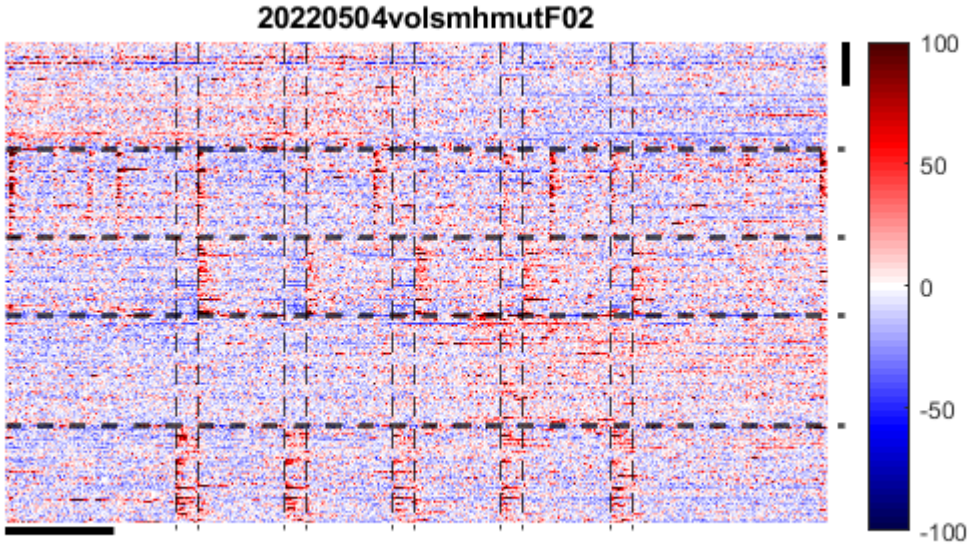
Opening file: 20220408volsmhmutF03Results_CLUSTERING.mat



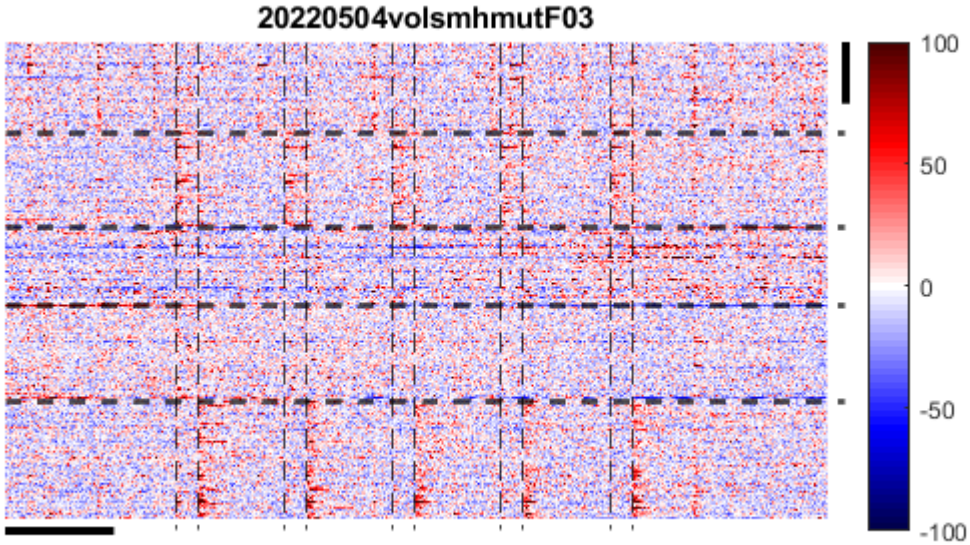
Opening file: 20220504volsmhmutF01Results_CLUSTERING.mat



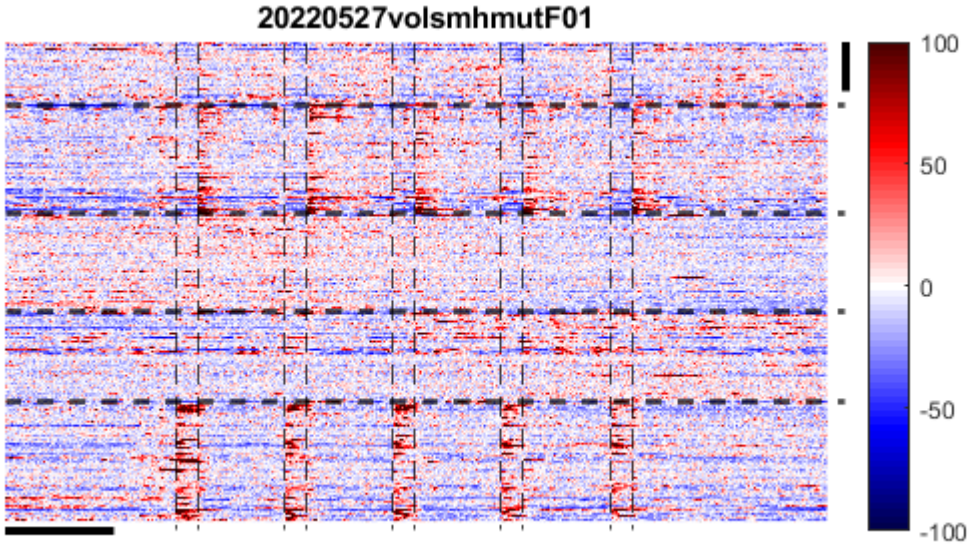
Opening file: 20220504volsmhmutF02Results_CLUSTERING.mat



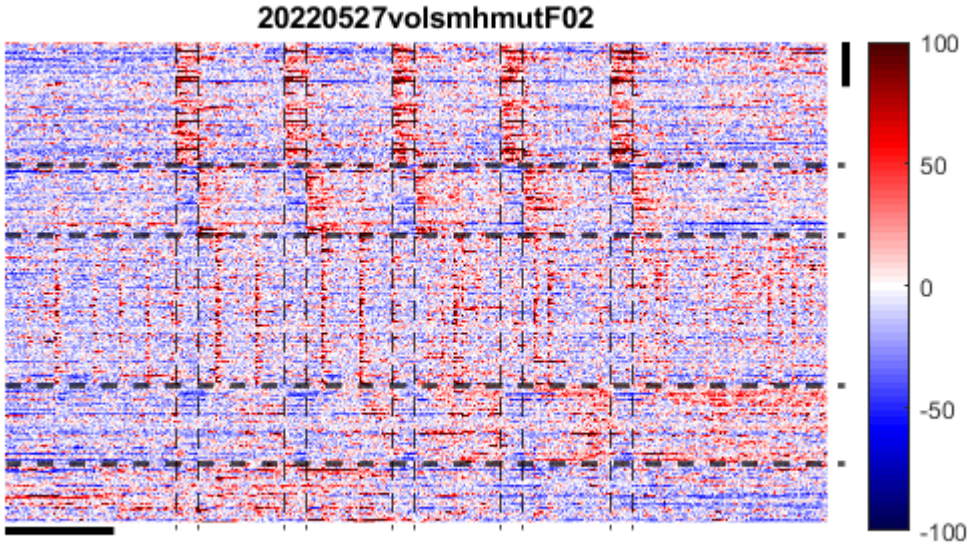
Opening file: 20220504volsmhmutF03Results_CLUSTERING.mat



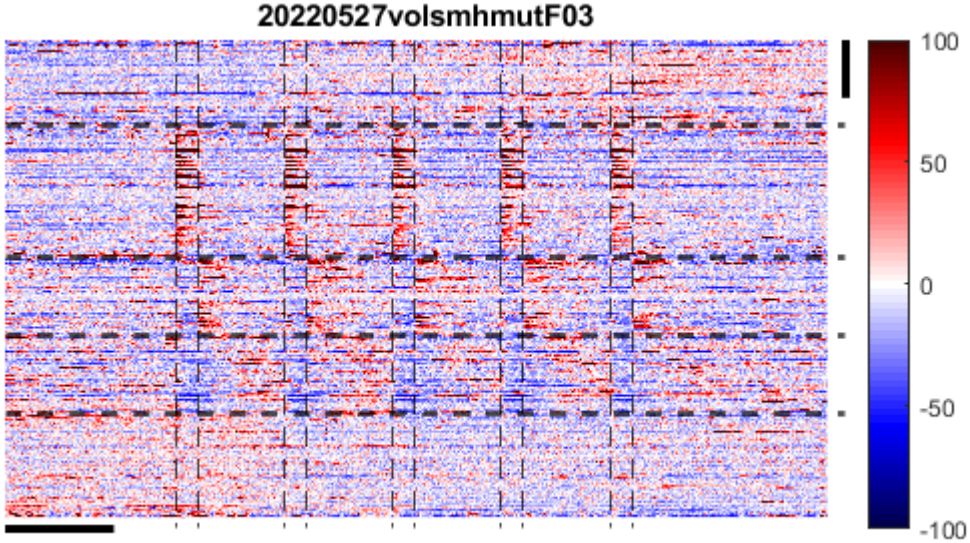
Opening file: 20220527volsmhmutF01Results_CLUSTERING.mat



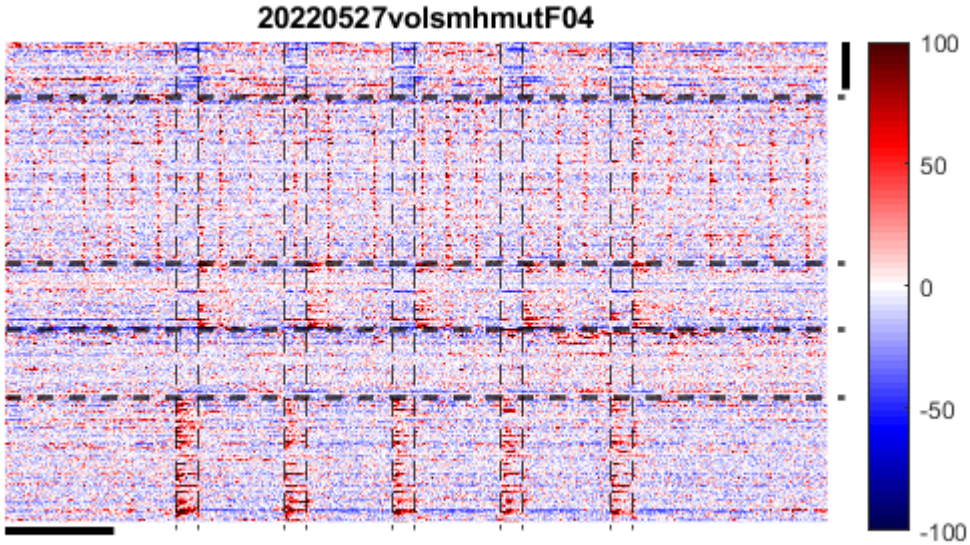
Opening file: 20220527volsmhmutF02Results_CLUSTERING.mat



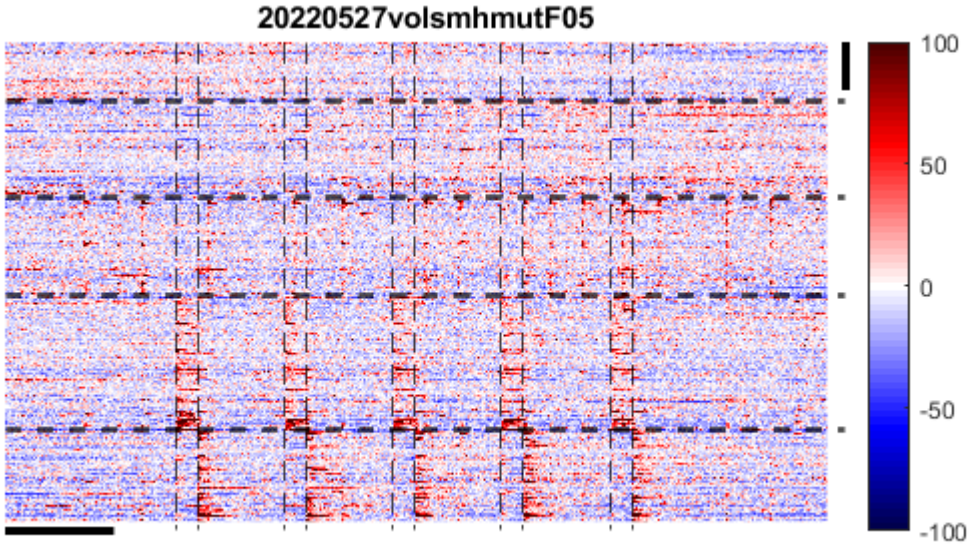
Opening file: 20220527volsmhmutF03Results_CLUSTERING.mat



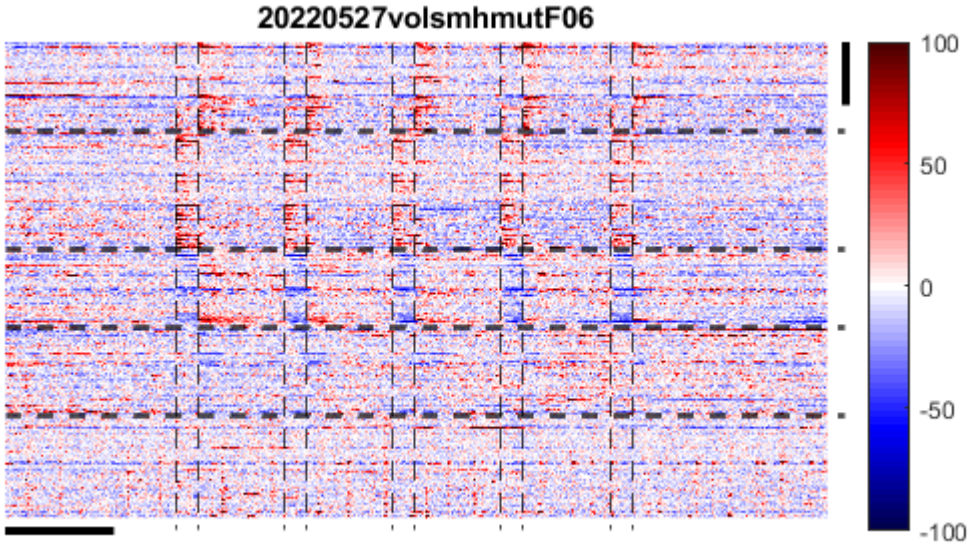
Opening file: 20220527volsmhmutF04Results_CLUSTERING.mat



Opening file: 20220527volsmhmutF05Results_CLUSTERING.mat



Opening file: 20220527volsmhmutF06Results_CLUSTERING.mat




```
function plot_heatmap(data, folderPath)
% HEATMAP-----------------------------------------------------------------
% plot the clustering with a bar for 5 min --------------------------------
% re-orderd the cluster so that cluster 1-2 is ON, cluster 3 is OFF, cluster
% 4 and 5 is not showing large activity
idx2=data.idx;
idx2(data.idx==4)=12;
idx2(data.idx==2)=13;
idx2(data.idx==5)=10;
idx2(data.idx==1)=11;
idx2(data.idx==3)=14;
% x scalebar is 5 min, y scalebar is 1000 neurons
cmap = seismic();
figure('Position', [100 100 600 300])
imagesc(sortrows(cat(2, idx2, data.DV_DFFaveragetime ))), hold on
colormap(cmap)
caxis([-100 100])
%colorbar
xline(data.cfg.TimeVector.ON,'k--','LineWidth',0.1)
xline(data.cfg.TimeVector.OFF,'k--','LineWidth',0.1)
yline(1+find(diff(sort(idx2))==1),'k--', 'LineWidth',2)
xlim([500 size(data.DV_DFFaveragetime,2)+200])
ylim([0 size(data.DV_DFFaveragetime,1)+200])
line([500 round(500+data.cfg.fps*60*5)], size(data.DV_DFFaveragetime,1)+[200 200],'Color','k','
line(size(data.DV_DFFaveragetime,2)+[200 200], [0 1000],'Color','k','LineWidth',3)
box off
axis off
ax=gca;
set(ax,'XTick',[])
set(ax,'YTick',[])
%set(ax,'TickDir','out')
title(data.metadata.name)
colorbar
saveas(gcf, [folderPath, data.metadata.name,'.png']);
```

```
end
```

```
function c=seismic()
c = [
    0.0, 0.0, 0.3;
    0.0, 0.0, 0.35;
    0.0, 0.0, 0.4;
    0.0, 0.0, 0.45;
    0.0, 0.0, 0.5;
    0.0, 0.0, 0.55;
    0.0, 0.0, 0.6;
    0.0, 0.0, 0.65;
    0.0, 0.0, 0.7;
    0.0, 0.0, 0.75;
    0.0, 0.0, 0.8;
    0.0, 0.0, 0.85;
    0.0, 0.0, 0.9;
    0.0, 0.0, 0.95;
    0.0, 0.0, 1.0;
    0.05, 0.05, 1.0;
    0.1, 0.1, 1.0;
    0.15, 0.15, 1.0;
    0.2, 0.2, 1.0;
    0.25, 0.25, 1.0;
    0.3, 0.3, 1.0;
    0.35, 0.35, 1.0;
    0.4, 0.4, 1.0;
    0.45, 0.45, 1.0;
    0.5, 0.5, 1.0;
    0.55, 0.55, 1.0;
    0.6, 0.6, 1.0;
    0.65, 0.65, 1.0;
    0.7, 0.7, 1.0;
    0.75, 0.75, 1.0;
    0.8, 0.8, 1.0;
    0.85, 0.85, 1.0;
    0.9, 0.9, 1.0;
    0.95, 0.95, 1.0;
    1.0, 1.0, 1.0;
    1.0, 0.95, 0.95;
    1.0, 0.9, 0.9;
    1.0, 0.85, 0.85;
    1.0, 0.8, 0.8;
    1.0, 0.75, 0.75;
    1.0, 0.7, 0.7;
    1.0, 0.65, 0.65;
    1.0, 0.6, 0.6;
    1.0, 0.55, 0.55;
    1.0, 0.5, 0.5;
    1.0, 0.45, 0.45;
    1.0, 0.4, 0.4;
    1.0, 0.35, 0.35;
```

```
    1.0, 0.3, 0.3;
    1.0, 0.25, 0.25;
    1.0, 0.2, 0.2;
    1.0, 0.15, 0.15;
    1.0, 0.1, 0.1;
    1.0, 0.05, 0.05;
    1.0, 0.0, 0.0;
    0.95, 0.0, 0.0;
    0.9, 0.0, 0.0;
    0.85, 0.0, 0.0;
    0.8, 0.0, 0.0;
    0.75, 0.0, 0.0;
    0.7, 0.0, 0.0;
    0.65, 0.0, 0.0;
    0.6, 0.0, 0.0;
    0.55, 0.0, 0.0;
    0.5, 0.0, 0.0;
    0.45, 0.0, 0.0;
    0.4, 0.0, 0.0;
    0.35, 0.0, 0.0;
    0.3, 0.0, 0.0;
];
end
```