Neural Imaging Data Analysis

edited DRA, 20210913

➢ **STEP 0: Preparation**

Before tracking neurons, it is recommended to create an experiment summary video to assess which animals responded to stimulation, moved between or during trials, and any that might be harder to track such as by overlapping other animals, leaving the field of view, etc.

An ImageJ/FIJI macro (Plugins → *ExperimentSummary_2FramesPerVideo*) creates a single video per experiment folder. It asks for the experiment folder name, finds all the raw .tif stacks, then asks for two frames numbers representing (1) **baseline** (e.g. frame 10... sometimes the first frames are a bit dimmer) and (2) **stimulus** (e.g. last frame # of stimulus is good). It extracts these 2 frames for each trial video and builds a separate video file. From this, you can identify which animals moved from trial to trial, which have neural responses, etc. to help plan which neurons to track using NeuroTracker.

It is recommended to copy a video frame to an editable document (e.g. PPT) and label each desired animal (sequentially from 0 to match NeuroTracker). Make note of any movements to help the tracking step, such as: "animal 4 moves to left at trial 5" so you know how to correct during the tracking process.

NeuroTracker (ImageJ)

➢ **STEP 1: Tracking Neurons**

NeuroTracker is an ImageJ plugin that extracts neural data from recorded videos (TIFF stacks). It takes in user input to choose many neuron positions and threshold values, then loops through each time point updating the neuron position, calculating integrated intensity (pixels within a recording box), background, and other data such as the threshold value and any user flags. The output is a text file for each neuron and trial, containing data (csv format) with one row per timepoint. The experiment folder contains the following files:

<u>NeuroTracker data files:</u>

"stream_[yyyy-mm-dd-hh-mm]_mov001_Pos.txt"

This file contains the position information recorded by the user during neuron selection. For example:

```
x 597.0 y 195.0 a 0 t 796.0 f 0 g 1
x 639.0 y 206.0 a 1 t 796.0 f 0 g 1
```

contains two animal/neuron positions: animal (a) 0 with neuron centroid (x,y) at (597,195) position, threshold (t) 796, and… what are f? g?

The file "**initialPos.txt"** is the same as the most recent *_Pos.txt file

"stream_[yyyy-mm-dd-hh-mm]_mov001.an0.txt"
This file contains the data for animal 0. For example:

```
Slice,xc,yc,intdens,intsub,bgmedian,maxint,area,x,y,sqintdens,sqin
tsub,sqarea,threshold,animal,redFlag,useTracking
1,592.630,196.396,7633.000,3905.000,466.000,1140.000,8.000,592.625
,196.375,23583.000,6807.000,36.000,752,0,0,1
2,592.622,196.400,7836.000,3948.000,486.000,1241.000,8.000,592.625
,196.375,24616.000,7120.000,36.000,752,0,0,1
...
```

Data are:
slice: time point number (integer)
xc,yc: centroid position in pixels
intdens: integrated density of thresholded neuron
intsub: intdens minus area * background median (i.e. sum of thresholded pixels, with background subtracted from each pixel
bgmedian: median pixel value of an annulus surrounding the neuron
maxint: maximum pixel value in thresholded area
area: size of thresholded area (pix^2)
x,y: centroid position in pixels
sqintdens: integrated density of all pixels within the designated square
sqintsub: sqintdens minus square area * background median (i.e. sum of thresholded pixels, with background subtracted from each pixel
sqarea: square area (pix^2)
threshold: threshold value
animal: animal number
redFlag: user-initiated flag to exclude data
useTracking: flag to use automatic neuron tracking frame-to-frame

NeuroTrackerSummary (MATLAB)

➤ **STEP 2: Data Validation**

NeuroTrackerSummary is a MATLAB script that reads in neural data from all NeuroTracker files (.txt files) within a folder. It cleans up the initial data and builds matrices and vectors that allow plotting of neural fluorescence over time, F(t), baseline fluorescence (F0), peak

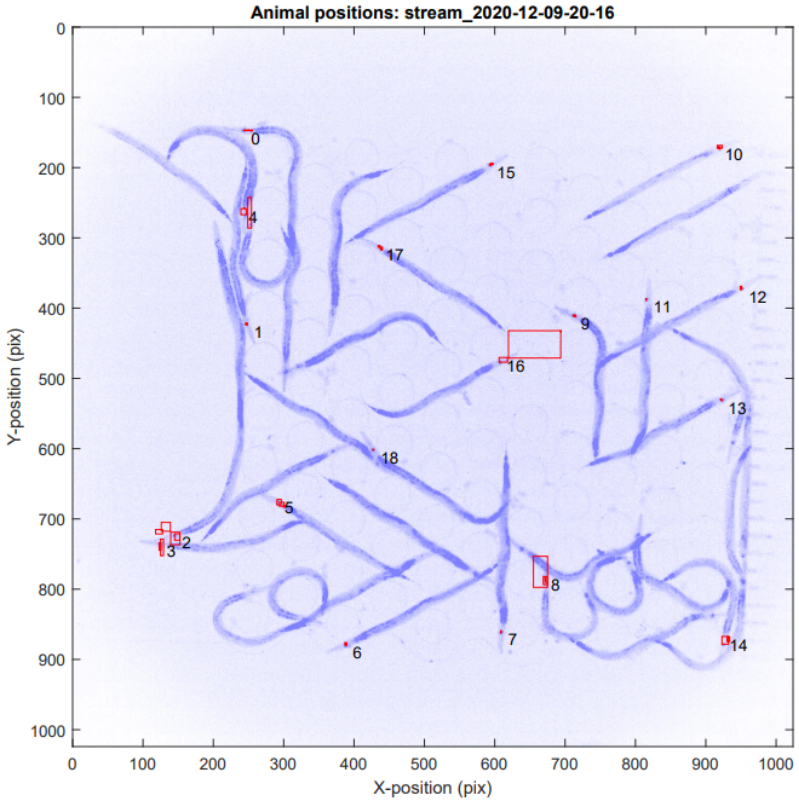
response, etc. The dataBrowseS function allows exploration of the data by sorting and selecting particular groups such as animals and trials.

The script first reads in all .txt files into the SmDat structure, including the time-varying data contained within, and animal, experiment, and time data from the file names.

It then saves several summary PDFs, including:

[base]_AnimalPos.pdf

The first image of the TIFF stack is overlaid with red boxes indicating the extent of movement during the recording trial. Labels correspond to the animal numbers. *[Check for correct head/tail, note movement if unexpected.]*



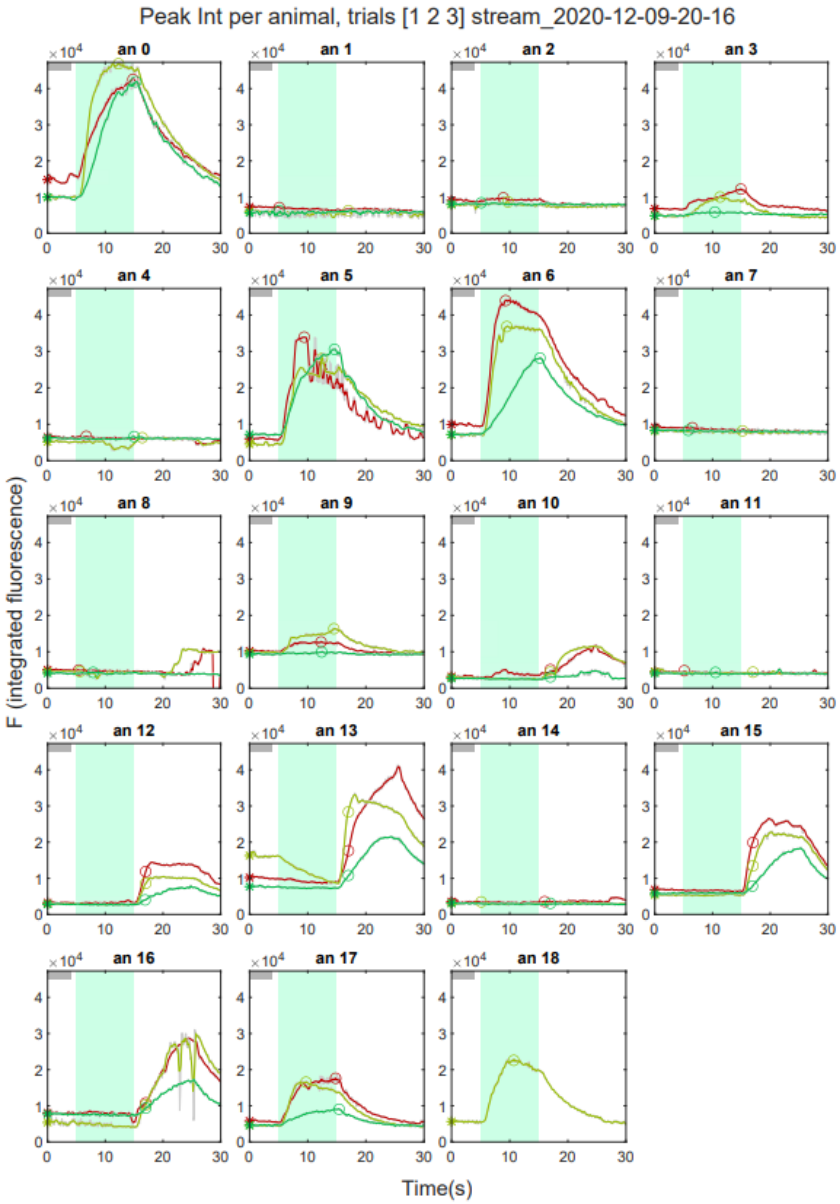
[base]_PeakInt_perAnimal.pdf

An array of plots show fluorescence F(t) over time t for each animal (an).

The baseline fluorescence F0 is indicated by * at time t=0, calculated as mean F during the baseline_t time window (gray bar at top).

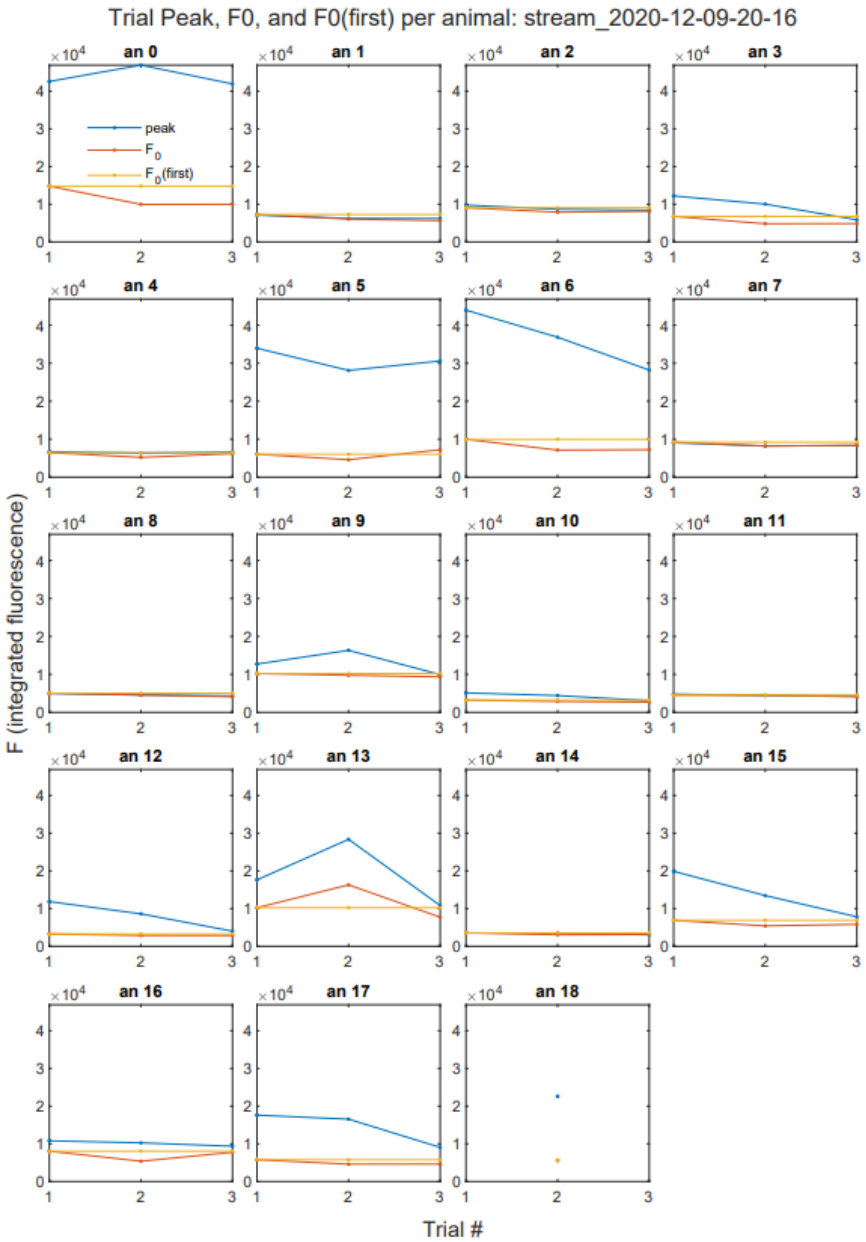
The stimulus window is highlighted green as defined by stimulus_t time window. Peak F(t) is determined as max value within stimulus region (or 2 s after).

[Check for correct baseline and peak detection, note unusual responses, such as animals 10,12,13,15, etc that respond after stimulus.]



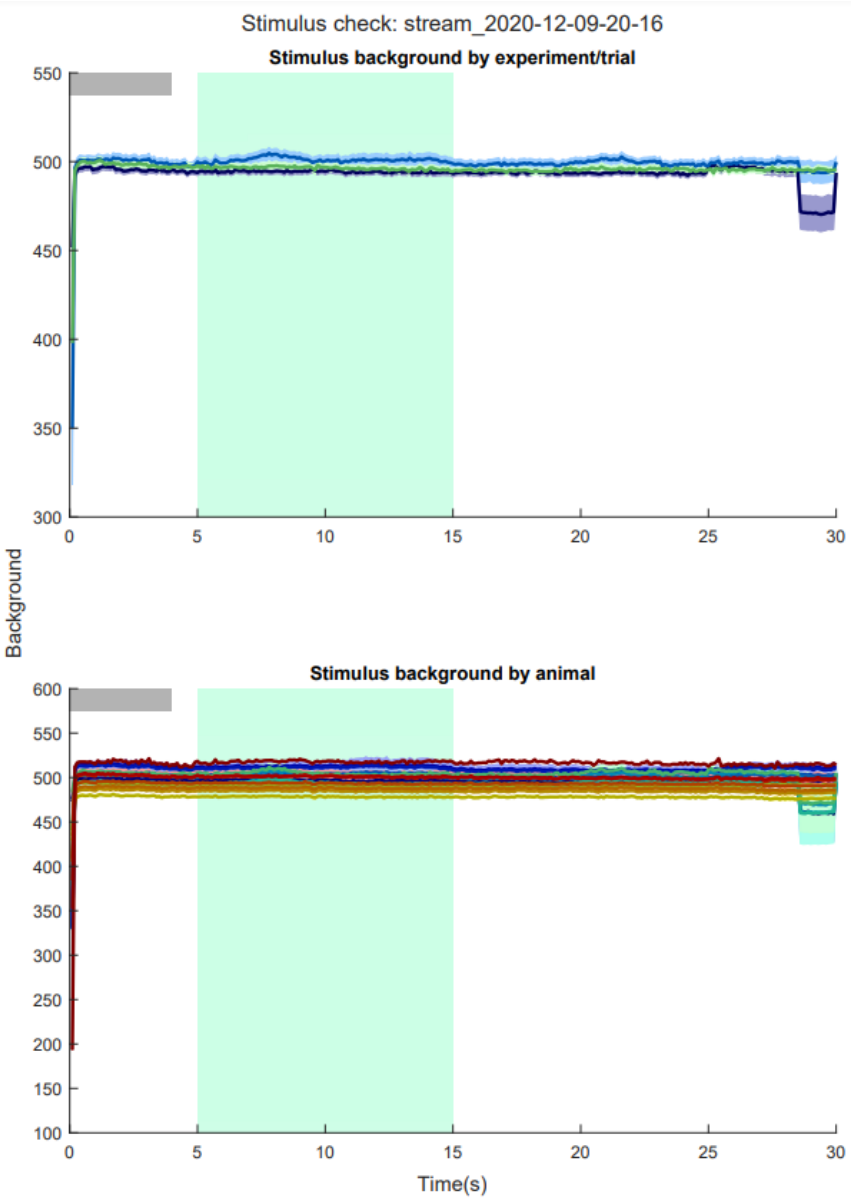
[base]_Peak_F0_perAnimalperTrial.pdf

An array of plots show raw, non-normalized fluorescence peak F, F0, and F0 of the first trial plotted across trials for each animal. This plot indicates trial-to-trial variation in baseline F0 and peak values. Any change in F0 (red) might indicate tracking error or shift in baseline, such as when F has not yet settled to baseline between rapid repeats and when a non-stimulus perturbation has elevated or suppressed the baseline (such as by electrical stimulation or injury). It is expected that peak F declines over trial # due to adaptation. *[Check for unexpected baseline and peak shifts, and use this information to properly normalize the F(t) data, such as to each trial's F0 or to the first trial F0.]*



[base]_StimulusCheck.pdf

Plots of background median intensity report stimulus on- and offset (if dye is included) and other experimental or tracking errors (such as fluid loss, bubbles, tracking shift to other animals). Upper plot averages all animals per trial; lower plot averages all trials per animal. Deviations from expectations may indicate an experiment failure or animals to be excluded from analysis, such as due to overcrowding and poor stimulation.



➤ **STEP 3: Data Analysis**

Make note of any animals or experiments to be excluded due to experiment error (e.g., incorrect stimulus timing) or tracking error.

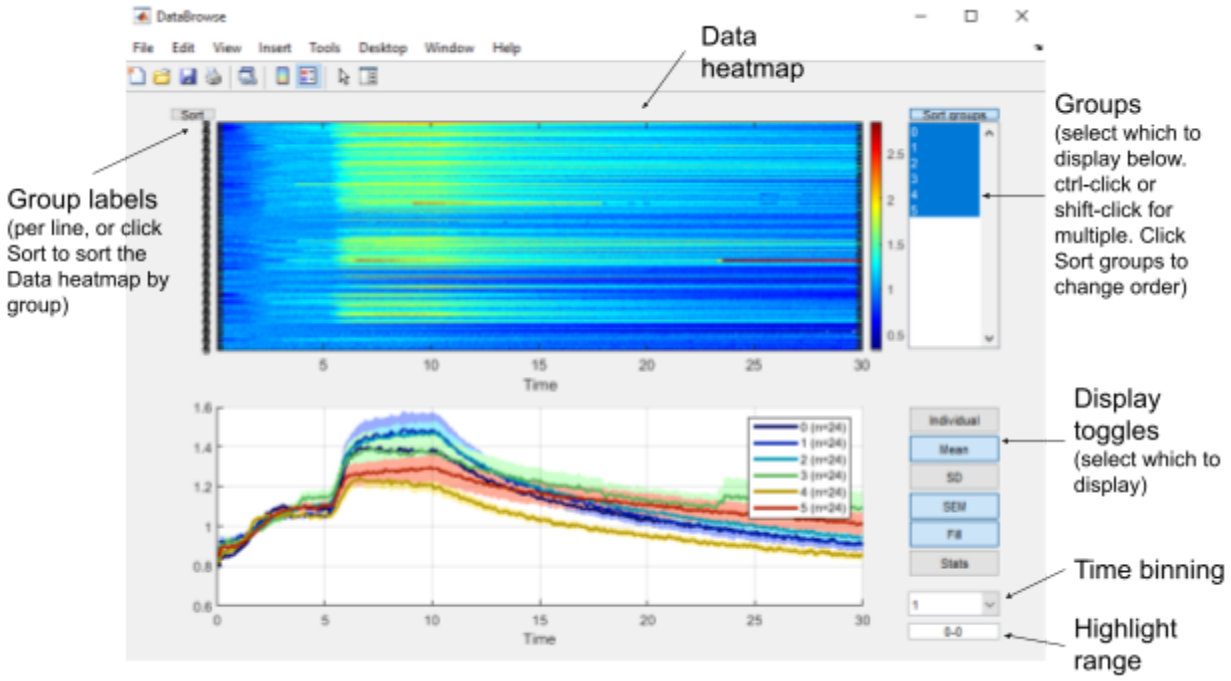
Use **databrowse** function to reduce data as desired and appropriate. Examples are shown below.

Usage of databrowseS:

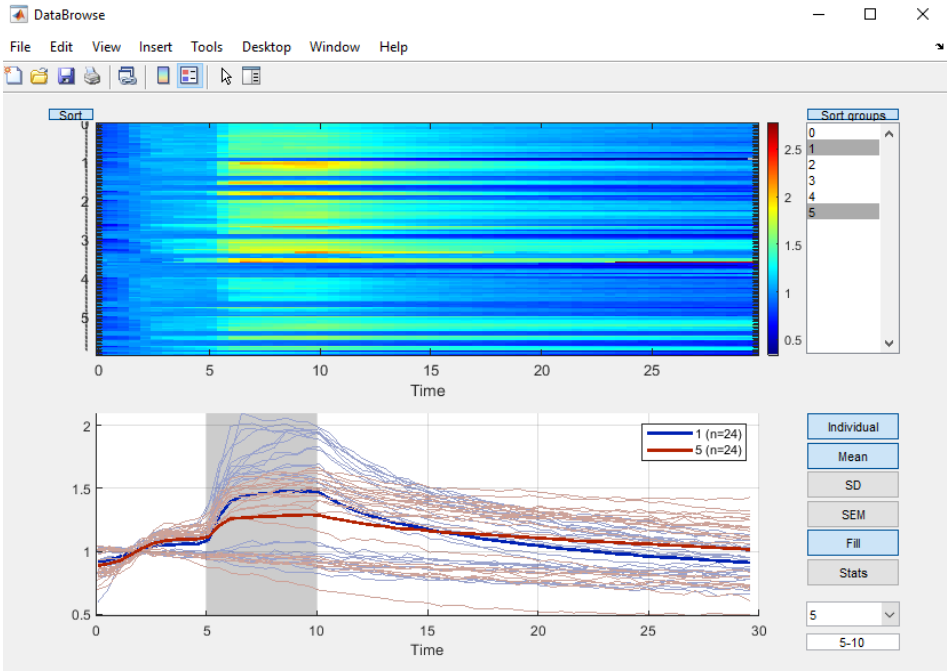
```
function handle = databrowseS(mat,x,ygroup,defaultsettings)
%
%          mat: m-by-n data matrix
%            x: 1-by-n optional labels for x-axis
%       ygroup: m-by-1 list of group nnames for each row in mat
%                 (can be numeric or cell)
%
%    defaultsettings: cell array containing values of all user objects
%           e.g. {1:length(groups),0,1,0,1,1,1,0}
%           (1) groups to display, (2) individual traces, (3) mean,
%           (4) SD, (5) SEM, (6) shaded error, (7) x-binning,
%           (8) show stats, (9) sort group order, (10) sort matrix rows
%
%       Note: can transfer figure elsewhere by:
%             h = databrowse(...); copyobj(get(h(1),'Children'),gca);
```

example:
databrowseS(AllSqIntNorm',t,animal)

matrix is AllSqIntNorm [300 x 144]
group is animal number 0-5

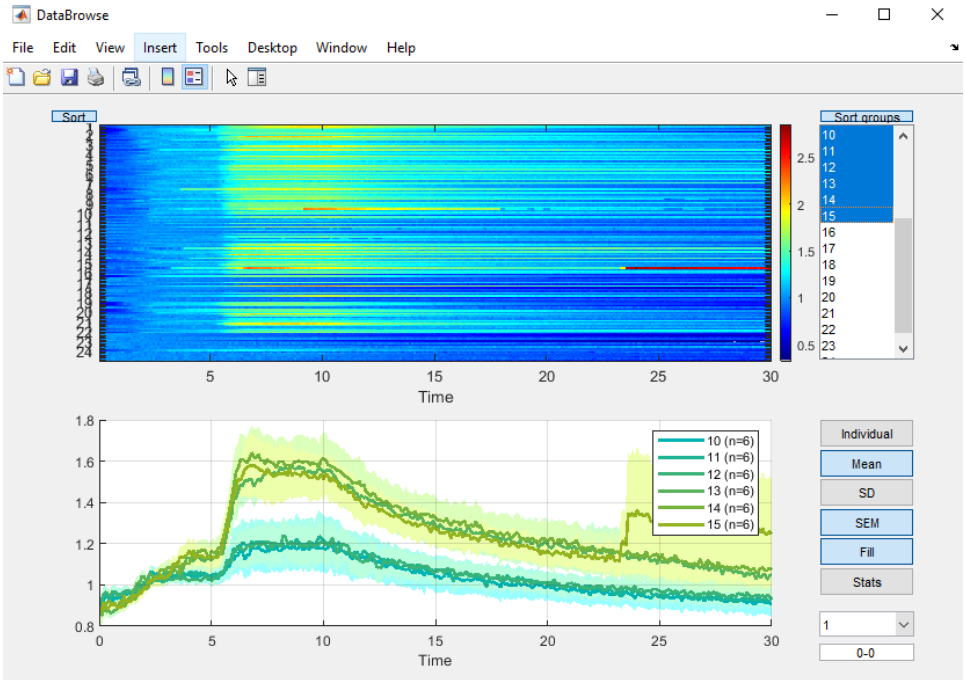


Same plot highlighting animals 1 & 5



Now group by experiment/trial:

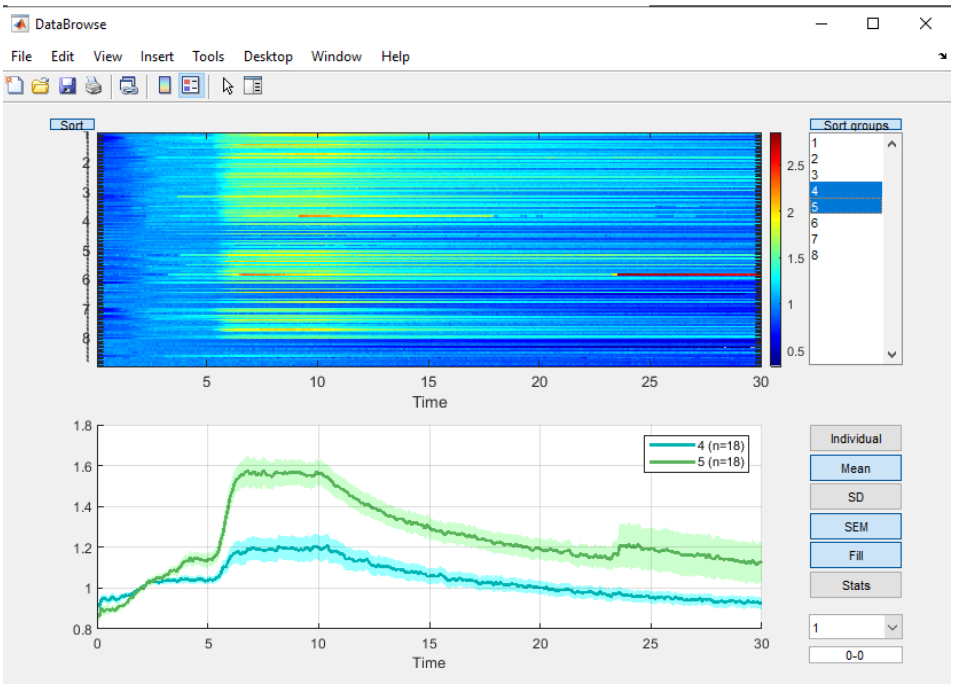
databrowseS(AllSqIntNorm',t,exp)



The heatmap shows something different for experiment repeats 10-12 and 13-15. highlighting these groups makes that clear. To group these together, define a new experiment grouping, such as exp 1-3 = expset 1, exp 4-5 = expset 2, etc.

expset = ceil(exp/3);
databrowseS(AllSqIntNorm',t,expset)

Now the 3 trials for each set are combined:

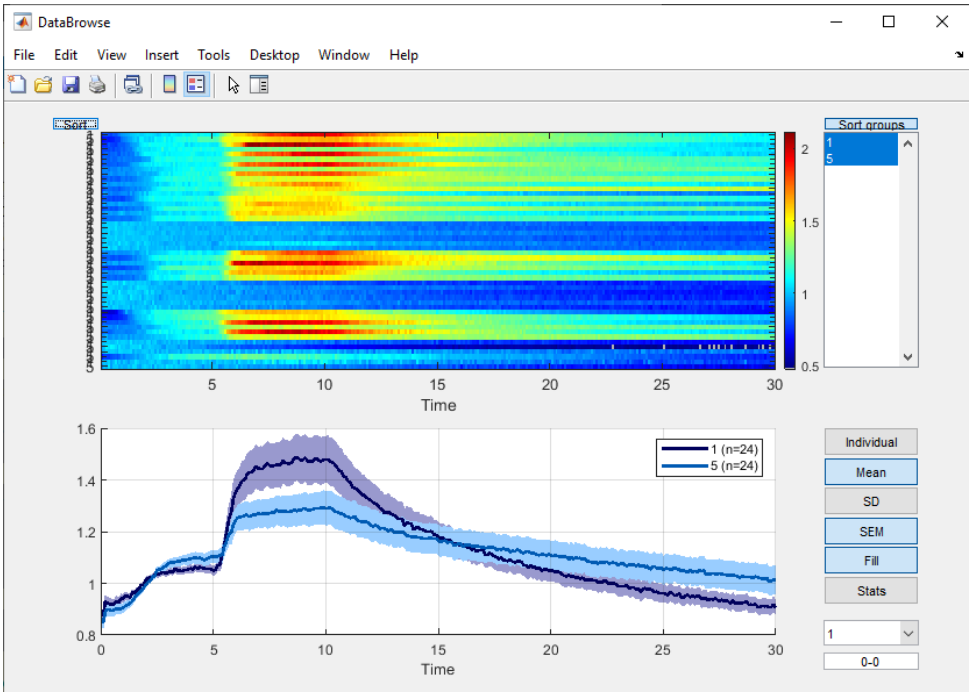


Data subsets

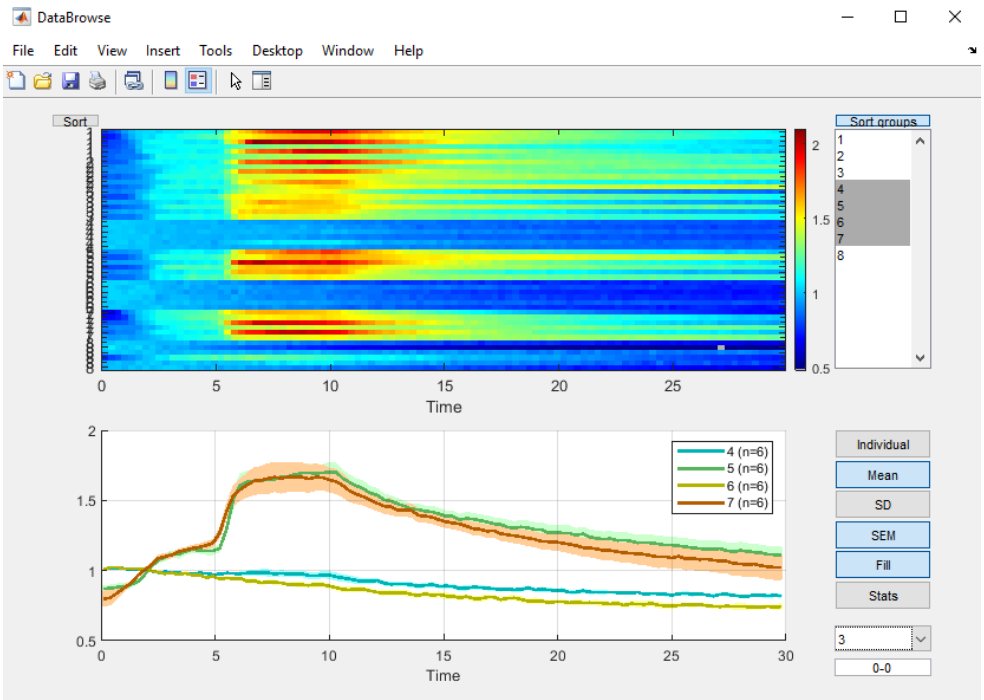
To explore subsets of data, such as specific experiments or animals, a good strategy is to find the trials corresponding to the desired groups and then databrowse again. For example in this dataset, animals 1 and 5 react similarly. Let's explore and perhaps group them together.

idx = find(animal == 1 | animal == 5);
databrowseS(AllSqIntNorm(:,idx)',t,animal(idx))

^^note that we now send a matrix with only the "idx" list of rows. We also need to pass the corresponding group only of the "idx" rows.

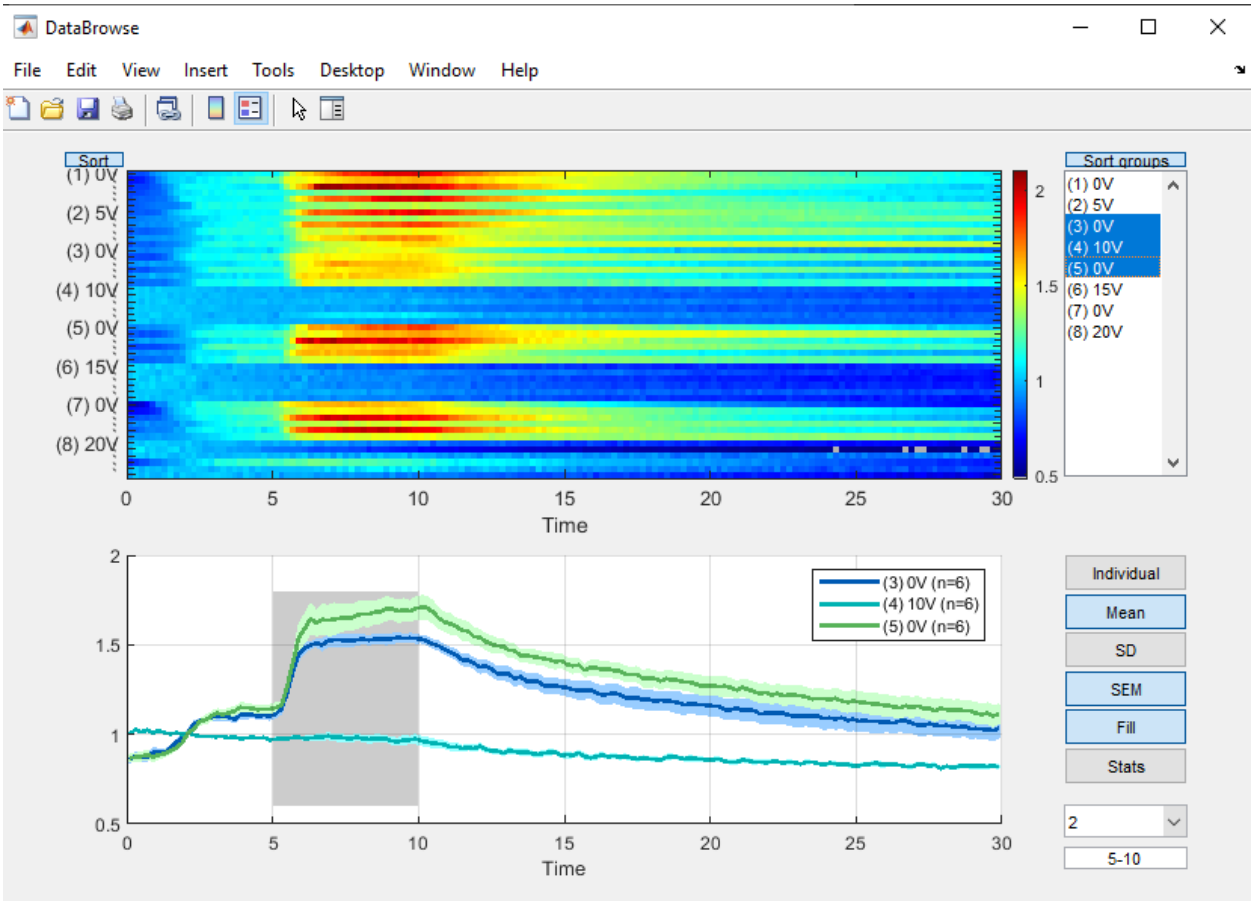


Similarly grouping just these two animals by expert:
databrowseS(AllSqIntNorm(:,idx)',t,expset(idx))



Let's make the groups more readable:
Define some labels in a cell array. Make sure that the array index corresponds to the group number (starting with 1). Then pass this label array to databrowse, with the appropriate grouping.

DBS_condition = {'(1) 0V','(2) 5V','(3) 0V','(4) 10V','(5) 0V','(6) 15V','(7) 0V','(8) 20V'};
databrowseS(AllSqIntNorm(:,idx)',t,DBS_condition(expset(idx)))



In summary, this plot shows average response from two animals (from idx, #1 and #5), three repeated trials (from expset) each, so n=6 per curve.

hilite([5,10;15,20])