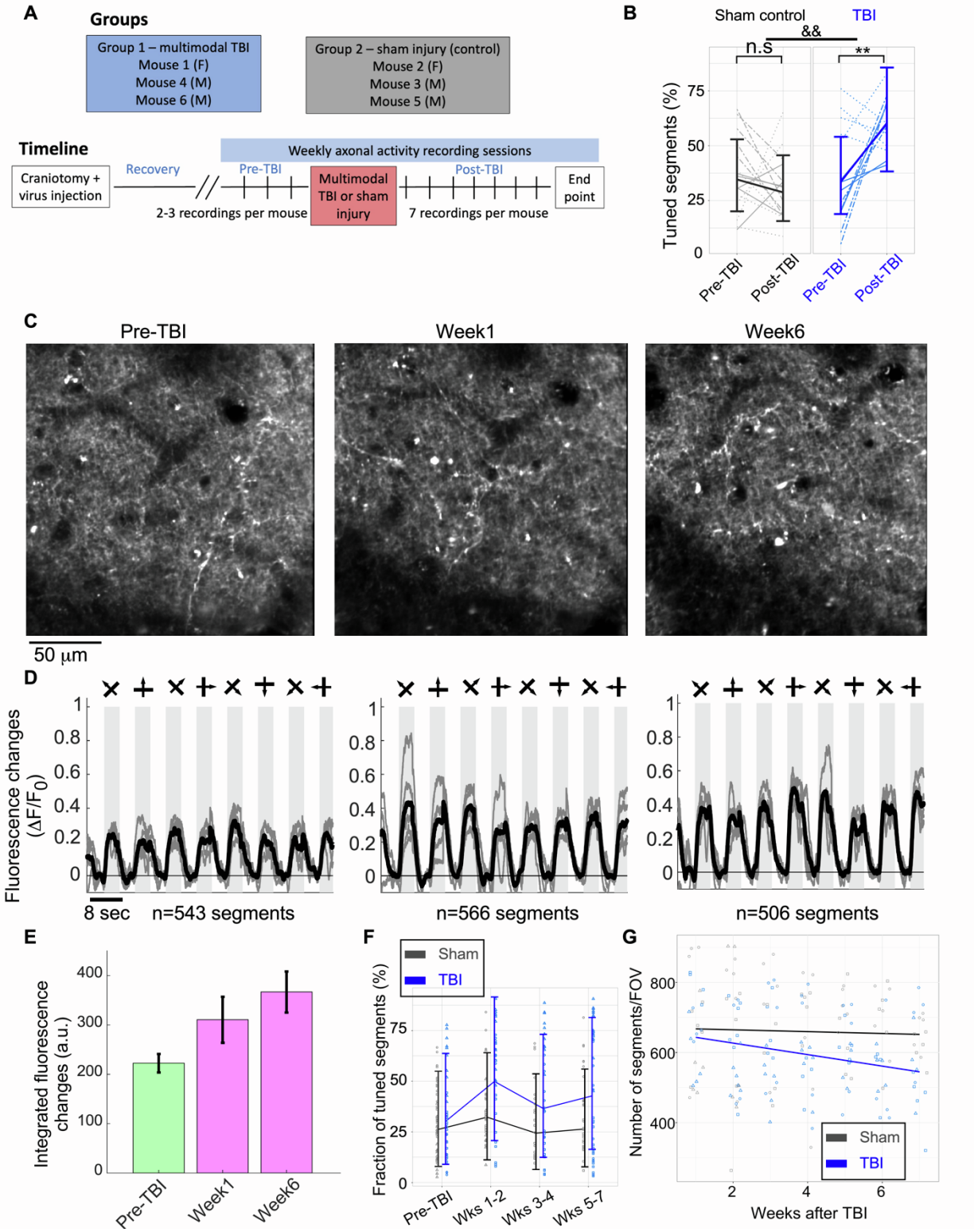
Cell Reports Methods, Volume *3*

Supplemental information

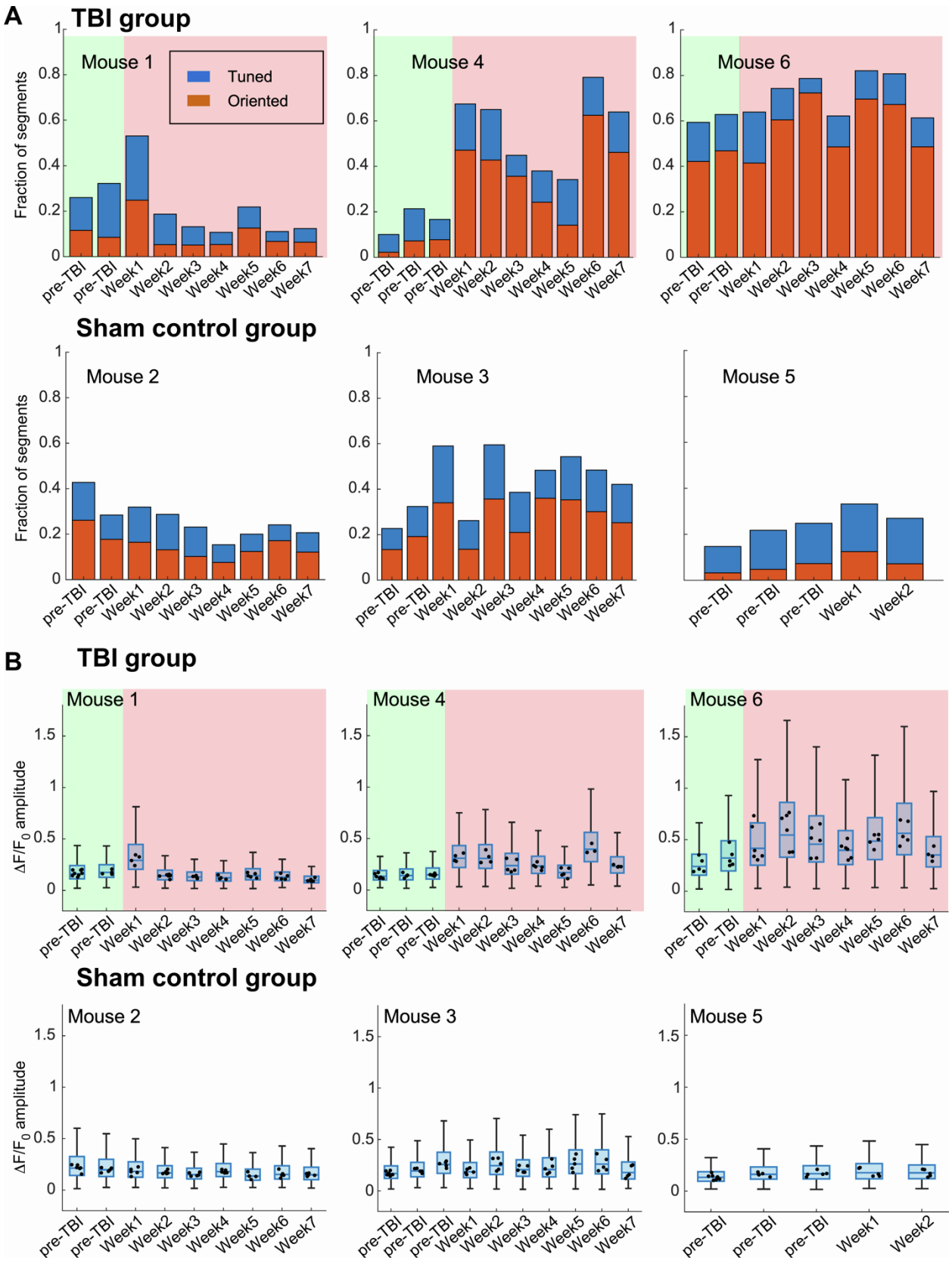
Longitudinal *in vivo* monitoring of axonal degeneration after brain injury

Sergiy Chornyy, Julie A. Borovicka, Davina Patel, Min-Kyoo Shin, Edwin Vázquez-Rosa, Emiko Miller, Brigid Wilson, Andrew A. Pieper, and Hod Dana

Supplementary information



Supplementary Figure 1. Experimental design and additional data related to STAR Methods and Main Figs. 1-2. A. Six male and female mice (M and F, respectively) were randomly divided into 2 groups: TBI and sham injury (control). All mice were implanted with cranial windows and injected with GCaMP6s-axon into their dLGN. We recorded 2-3 sessions of baseline visual stimulated activity before exposing mice in the TBI group to multimodal TBI. Mice in the sham injury group were anesthetized and placed inside the multimodal TBI device without activating the overpressure chamber. We continued recording axonal activity for seven weeks after injury. **B.** Short-term increase in axonal tuning after multimodal TBI. The mean fraction of axonal segments that were significantly tuned ($P<0.01$, ANOVA test for tuning of segments, individual segment data not shown; see Methods for details) to the presented visual stimulus showed a significant group-by-time interaction between the TBI and sham control groups ($P=0.0017$, F-test). For the TBI group, the fraction was significantly increased by 74% (blue; $P=0.002$, Holm-adjusted) in the days before and after TBI, but not for the sham control group (gray). &&, $p<0.01$ for group-by-time interaction; **, $p<0.01$ for within-group comparison; n.s., not significant). For each experimental group, different lines (solid, dotted, dash-dotted) show data from different mice, error bars show the 95% confidence interval for the linear mixed-effects model, and the model mean values are connected with lines. **C.** Repeated recordings from the same FOV. Example mean images of the same FOV (mouse 6, TBI group) recorded 1 day before TBI (left), 4 days after TBI (middle), and 6 weeks after TBI (right). Activity movies from these recordings are shown in Supplementary Video 1. **D.** Mean $\Delta F/F_0$ traces across all detected segments for each FOV shown in **C**. Single trials (gray) and averages of five trials (black) are overlaid. Eight grating motion directions are indicated by arrows as shown above the traces. **E**. Sum of the $\Delta F/F_0$ traces in **D** shows an increase in response amplitudes after TBI (mean±s.d. for the 5 trials). **F.** Increase in the fraction of tuned segments for the TBI group. The percentage of tuned segments ($P<0.01$, one-way ANOVA for testing the tuning of segments to the stimulus; data for individual segments not shown; see Methods for details) was non-significantly increased for the TBI group over the weeks after injury and no increase was identified for the sham control group (no significant group-by-time interaction was detected, $P=0.092$, F-test). **G.** Gradual decrease in the number of detected segments after TBI. The number of segments per FOV showed a decrease in the TBI group (blue), with a model-estimated decrease of 15.2% from Week 1 to Week 7. A small decrease was identified in the sham control group (gray). These changes are consistent with reported findings on axonal degeneration in this multimodal TBI model. We note that the changes in our study were not significant for group and time interaction ($P=0.108$, F-test), suggesting that either more mice per group and/or longer recording periods are required to identify significant differences. For each group, data from single mice are represented by different markers (square, triangle, circle), and the model means are connected with lines.


A
TBI group
Mouse 1
Tuned
Oriented
Mouse 4
Mouse 6
Fraction of segments
Sham control group
Mouse 2
Mouse 3
Mouse 5
pre-TBI
Week1
Week2
Week3
Week4
Week5
Week6
Week7
B
TBI group
Mouse 1
Mouse 4
Mouse 6
ΔF/F0 amplitude
Sham control group
Mouse 2
Mouse 3
Mouse 5

Supplementary Figure 2. Longitudinal monitoring of tuning, orientation, and response amplitudes for individual mice before and after multimodal TBI, related to Main Fig. 2. A. The fractions of tuned and oriented segments (blue and orange bars, respectively) from all recorded segments and all mice are shown for the multimodal TBI group (upper row), and sham injury group (bottom row). Green and red backgrounds indicate the pre- and post-TBI periods, respectively, for the TBI group. Data from all recorded FOVs on the same date were compiled to calculate the respective fractions. Note that data from mouse 5 were acquired for only 5 recording dates. **B.** Single-mouse $\Delta F/F_0$ response amplitudes are shown for the multimodal TBI (upper row), and sham control (bottom row) groups. Each box represents the 25-75 percentile range of the measured $\Delta F/F_0$ response amplitude from all recorded segments in a specific recording session. The whisker spans the smallest among the range of the entire data set, or 1.5 times the interquartile range. For each recording date, the median values from all recorded FOVs are overlaid on the boxplot data (black dots).

Methods S1. Matlab scripts and functions, which were used for the data analysis. Related to STAR Methods 1.

```
%this script receives a Matlab file of raw fluorescence recording,
%optionally after registration to fix small movements during recording.
%This file is called "reg"
% The recording was done at 30fps, where first a grey display is shown for
% 120 frames, and then the stimulus (1 of 8 directions) appears for 121
% frames.
%The code receives the pixels that identify the location of different
%axonal segments from the output of Suite2P, which is saved in the same
%folder as the recording data for each recorded FOV.

nCond=8;
nTrial=5;
nImPerBlock=241;
BasePeriod=(101:120);
StimPeriod=(121:241);
fs=30; %frames/sec
relevant_frames=1:nImPerBlock*nCond*nTrial; %these are the frames when the drifting grating stim was presented
neuropil_corr=0.7;

%% multiple FOVs
subdirs = uipickfiles('Output','Struct');
len=length(subdirs);
ROI_pile=[];
for m=1:len
   if subdirs(m).isdir
      files1=dir([subdirs(m).name,'/reg_movie_30Hz*']);
      files2=dir([subdirs(m).name,'/Fall.mat']);

      load([subdirs(m).name,filesep,files1(1,1).name]); %reg file
      load([subdirs(m).name,filesep,files2(1,1).name]); %Fall file

      num_seg_in_FOV(m)=size(F,1);
      %% load segmentation from Suite2P file, and apply it to the reg file
      if size(reg,3)~=10000
         beep
         error('wrong number of frames in reg file, check data')
      end
      ROI_list=struct('pixels',[],'fmean',[]);
      reg_mod=reshape(reg,[],size(reg,3));
      Fmean=zeros(size(F,1),size(reg,3));
```

```
        for i=1:size(Fmean,1)
            xpix=[];
            ypix=[];
            xpix=[xpix stat{i}.xpix];
            ypix=[ypix stat{i}.ypix];
            if sum(xpix==0)|sum(ypix==0) %if a pixels is at the edge
                ROI_list(i).pixels=sub2ind([size(reg,1) size(reg,2)],ypix+1, xpix+1);
            else
                ROI_list(i).pixels=sub2ind([size(reg,1) size(reg,2)],ypix, xpix);
            end
        end
        ROI_list=measure_neuropile_axon_suite2p(ROI_list,mean(reg,3));
        for i=1:size(F,1)
            ROI_list(i).fmean=mean(reg_mod(ROI_list(i).pixels,:),1);
            ROI_list(i).fmean_neuropile=mean(reg_mod(ROI_list(i).pixels_neuropile,:),1);
        end
    end
    ROI_pile=[ROI_pile ROI_list];
end

for i=1:length(ROI_pile)
    Fmean(i,:)=ROI_pile(i).fmean-neuropil_corr*ROI_pile(i).fmean_neuropile;
end
Fmean=Fmean(:,relevant_frames);

Fmean_org=Fmean;

nROI=size(Fmean,1);
Fmean=reshape(Fmean',[nImPerBlock,nCond,nTrial,nROI]); %transpose Fmean to fit the reshape format
df_f=rawf2df_f_init_f0(Fmean,BasePeriod);
df_f=reshape(df_f,[nImPerBlock,nCond,nTrial,nROI]);%transpose df/f to fit the reshape format

resp_avg=squeeze(mean(mean(df_f(StimPeriod,:,:,:),1),3));   % nstim x nROI
bestresp=max(resp_avg);
minresp=min(resp_avg);
resp_amplitude=bestresp-minresp;
for i=1:length(bestresp)
    ind=find(bestresp(i)==resp_avg(:,i));
    trace_mean=mean(squeeze(df_f(StimPeriod,ind,:,i)),2);
    trace_mean_sorted=sort(trace_mean,'descend');
    resp_peak(i)=mean(trace_mean_sorted(1:round(0.25*length(trace_mean_sorted))));
end
[IsRespond,IsOrient,p_resp,p_orient]=testResp(Fmean,BasePeriod,StimPeriod,0.05);
IsRespond=IsRespond&(bestresp>0.05);
IsOrient=IsOrient&(bestresp>0.05);
responsive_seg_num=sum(IsRespond)
oriented_seg_num=sum(IsOrient)
total_num_segments=length(IsRespond)

resp_amp_percentile=[myprctile(resp_amplitude,25) myprctile(resp_amplitude,50) myprctile(resp_amplitude,75)]
%correction for drifting of baseline which makes negative df/f

%%  fit tuning
ind=find(IsRespond);
angle=[45,90,135,180,225,270,315,360];
```

```
para_list=[];
for i=1:length(ind)
   para=fit_tuning(resp_avg(1:8,ind(i)),angle);
   para_list=[para_list,para];
end

save Combined_analysis_20201124
```

2.

```
%this script piles the data from individual FOVs to get piled data for a
%specific recording date.

close all;clear;clc
%%piled data
load('Combined_analysis_20211105.mat', 'num_seg_in_FOV', 'IsRespond', 'IsOrient','resp_amplitude','para_list')
total_num_segments=length(IsRespond)
resp_amp_median=median(resp_amplitude)
responsive_seg_num=sum(IsRespond)
oriented_seg_num=sum(IsOrient)
fraction_responsive=responsive_seg_num/total_num_segments
fraction_oriented=oriented_seg_num/total_num_segments
OSI_median=median([para_list.OSI])
DSI_median=median([para_list.DSI])

%% individual FOV data

switch length(num_seg_in_FOV)
    case 3
         median_resp_amp_per_FOV=[median(resp_amplitude(1:num_seg_in_FOV(1)))
median(resp_amplitude(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
         median(resp_amplitude(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))]

       IsResp_per_FOV=[sum(IsRespond(1:num_seg_in_FOV(1)))
sum(IsRespond(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
         sum(IsRespond(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))]

       IsOrient_per_FOV=[sum(IsOrient(1:num_seg_in_FOV(1)))
sum(IsOrient(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
         sum(IsOrient(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))]

        case 4
       median_resp_amp_per_FOV=[median(resp_amplitude(1:num_seg_in_FOV(1)))
median(resp_amplitude(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
         median(resp_amplitude(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
         median(resp_amplitude(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))]

       IsResp_per_FOV=[sum(IsRespond(1:num_seg_in_FOV(1)))
sum(IsRespond(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
         sum(IsRespond(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
         sum(IsRespond(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))]

       IsOrient_per_FOV=[sum(IsOrient(1:num_seg_in_FOV(1)))
sum(IsOrient(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
         sum(IsOrient(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
```

```
        sum(IsOrient(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))]

    case 5
      median_resp_amp_per_FOV=[median(resp_amplitude(1:num_seg_in_FOV(1)))
median(resp_amplitude(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:4)):end))]

      IsResp_per_FOV=[sum(IsRespond(1:num_seg_in_FOV(1)))
sum(IsRespond(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:4)):end))]

      IsOrient_per_FOV=[sum(IsOrient(1:num_seg_in_FOV(1)))
sum(IsOrient(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:4)):end))]

    case 6
      median_resp_amp_per_FOV=[median(resp_amplitude(1:num_seg_in_FOV(1)))
median(resp_amplitude(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:5)):end))]

      IsResp_per_FOV=[sum(IsRespond(1:num_seg_in_FOV(1)))
sum(IsRespond(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:5)):end))]

      IsOrient_per_FOV=[sum(IsOrient(1:num_seg_in_FOV(1)))
sum(IsOrient(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:5)):end))]

    case 7
      median_resp_amp_per_FOV=[median(resp_amplitude(1:num_seg_in_FOV(1)))
median(resp_amplitude(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:5)):sum(num_seg_in_FOV(1:6))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:6)):end))]

      IsResp_per_FOV=[sum(IsRespond(1:num_seg_in_FOV(1)))
sum(IsRespond(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
```

```
        sum(IsRespond(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:5)):sum(num_seg_in_FOV(1:6))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:6)):end))]

    IsOrient_per_FOV=[sum(IsOrient(1:num_seg_in_FOV(1)))
sum(IsOrient(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:5)):sum(num_seg_in_FOV(1:6))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:6)):end))]

    case 8
    median_resp_amp_per_FOV=[median(resp_amplitude(1:num_seg_in_FOV(1)))
median(resp_amplitude(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:5)):sum(num_seg_in_FOV(1:6))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:6)):sum(num_seg_in_FOV(1:7))))...
        median(resp_amplitude(1+sum(num_seg_in_FOV(1:7)):end))]

    IsResp_per_FOV=[sum(IsRespond(1:num_seg_in_FOV(1)))
sum(IsRespond(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:5)):sum(num_seg_in_FOV(1:6))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:6)):sum(num_seg_in_FOV(1:7))))...
        sum(IsRespond(1+sum(num_seg_in_FOV(1:7)):end))]

    IsOrient_per_FOV=[sum(IsOrient(1:num_seg_in_FOV(1)))
sum(IsOrient(1+num_seg_in_FOV(1):sum(num_seg_in_FOV(1:2))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:2)):sum(num_seg_in_FOV(1:3))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:3)):sum(num_seg_in_FOV(1:4))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:4)):sum(num_seg_in_FOV(1:5))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:5)):sum(num_seg_in_FOV(1:6))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:6)):sum(num_seg_in_FOV(1:7))))...
        sum(IsOrient(1+sum(num_seg_in_FOV(1:7)):end))]
end
```

3.

```
function df_f=rawf2df_f_init_f0(f,f0_range)
% f2=reshape(f,121,[]);
ss=size(f);
if (length(ss)==2)
   ss(3)=1;
   ss(4)=1;
end
f0=zeros(ss);

for  i=1:ss(3)
   for j=1:ss(4)
      f0(:,1,i,j)=mean(f(f0_range,1,i,j));
      for k=2:ss(2)
         f0(:,k,i,j)=f0(:,1,i,j);
```

```
        end
    end
end

f0=reshape(f0,ss(1),[]);
f=reshape(f,ss(1),[]);

% f0=mean(f(f0_range,:));

% f0=repmat(f0,[ss(1),1]);
df_f=(f-f0)./f0;
df_f=reshape(df_f,ss);
```

4.

```
function [IsRespond,IsOrient,p_resp,p_orient]=testResp(fmean,BasePeriod,StimPeriod,threshold)
%this script runs an ANOVA test to find tuned and oriented segments
%fmean: 4D array nImPerBlock x nCond x nTrial x nROI

nImPerBlock=size(fmean,1);
nCond=size(fmean,2);
nTrial=size(fmean,3);
nROI=size(fmean,4);

IsRespond=false(1,nROI);
IsOrient=false(1,nROI);

%%  test for response,  ANOVA P<0.01

fbase=mean(fmean(BasePeriod,:,:,:));
fbase=reshape(fbase,[],nROI);åå
fresp=mean(fmean(StimPeriod,:,:,:));
fresp=reshape(fresp,[],nROI);
group=[zeros(nTrial*nCond,1);repmat((1:nCond)',[nTrial,1])];
ftestmatrix=[fbase;fresp];

for i=1:nROI
    p_resp(i)=anova1(ftestmatrix(:,i),group,'off');
    if p_resp(i)<threshold
        IsRespond(i)=true;
    else
        IsRespond(i)=false;
    end
end

%%  test for Orientation, ANOVA P<0.01
fresp=mean(fmean(StimPeriod,:,:,:));
fresp=reshape(fresp,[],nROI);
group=repmat((1:nCond)',[nTrial,1]);
for i=1:nROI
    p_orient(i)=anova1(fresp(:,i),group,'off');
    if (p_orient(i)<threshold) && (IsRespond(i)==1)
        IsOrient(i)=true;
    else
        IsOrient(i)=false;
    end
end
```