**Instructions of MATLAB scripts and Data used in in vivo ephysi analysis**

Ephys **raw** data for statistics in MATLAB format,Ephys data other than raw, MATLAB script written by Mai Iwasaki, Program from Multichannel Systems, Program from Dr. Rodrigo Quian Quiroga, Program from Dr. Maik Stüttgen, Script from an anonymous people.

Our original data of in vivo electrophysiology was saved in “.mcd” file, which the recording hardware company, multichannel systems (Reutlingen, Germany) developed. To make the file importable on MATLAB, “.mcd” file has to be converted to HDF5 file by multichannel data manager software.

<https://www.multichannelsystems.com/software/multi-channel-datamanager>

Next, to extract the data from HDF5 and reorganize it to be MATLAB friendly, please download McsMatlabDataTools.  
<https://github.com/multichannelsystems/McsMatlabDataTools>

The script “s1\_HDF5\_to\_WaveClus.m” extract the raw data from HDF5 using McsMatlabDataTools, and save the data for the next step, spike sorting by "wave clus". Modify “s1\_HDF5\_to\_WaveClus.m” to fit your settings, and run it.

Download Wave Clus for spike sorting, and add all the folders to the MATLAB path.

<https://github.com/csn-le/wave_clus>

In the file “set\_parameters.m”, we changed those parameters below otherwise used the default values.

param.sr = 25000;

par.w\_post = 40;

par.detection = 'both

Open the folder “TtrdV\_yy\_mm\_dd\_HH\_MM” made by “s1\_HDF5\_to\_WaveClus.m”.

Type in the matlab command window “Get\_spikes\_pol(1)” , “Do\_clustering\_pol(1)”, and then “wave\_clus”. On the appeared GUI, pick up the cluster (#1, 2, 3,,,) where you see the shape of spike, and find in which recording channel (a, b, c,,,) the spike is most prominently observed.

Download “mlib tool box” to analyze the spike after spike sorting by Wave Clus.

<https://www.mathworks.com/matlabcentral/fileexchange/37339-mlib-toolbox-for-analyzing-spike-data>

In “mcheck2” function, we modified this part;  
%% subplot 6 - stationarity: firing rate histogram over time, amplitudes over time

% w = get(subplot(446),'Position'); % somehow yields different positions

subplot('Position',[0.34264 0.54832 0.12 0.15554]) % left bottom width height

try

title('stability over time'),hold on

%[n,b] = hist(spxtimes,0:60:max(spxtimes));

[n,b] = hist(spxtimes,0:1:max(spxtimes));

%[ax,h1,h2] = plotyy(b/60,n,spxtimes/60,amps,'bar','plot');

[ax,h1,h2] = plotyy(b,n,spxtimes,amps,'bar','plot');

%pars = polyfit(spxtimes/60,amps,1);

pars = polyfit(spxtimes,amps,1);

set(gcf,'CurrentAxes',ax(2))

%line(spxtimes/60,pars(1)\*spxtimes/60+pars(2),'Color','r') % plot seems to destroy parent figure

line(spxtimes,pars(1)\*spxtimes+pars(2),'Color','r') % plot seems to destroy parent figure

set(h1,'LineStyle','-')

set(h2,'LineStyle','none','Marker','.','MarkerSize',3)

%set(get(ax(1),'Ylabel'),'String','spike count (60s bins)')

set(get(ax(1),'Ylabel'),'String','spike count (1s bins)')

set(get(ax(2),'Ylabel'),'String','ADC units')

%set(ax(1),'Xlim',[0 ceil(max(b/60))],'Ylim',[0 max(n)\*2])

set(ax(1),'Xlim',[0 ceil(max(b))],'Ylim',[0 max(n)\*2])

%set(ax(2),'Xlim',[0 ceil(max(b/60))],'Ylim',[min(amps)\*0.2 max(amps)\*1.1],'YColor','r')

set(ax(2),'Xlim',[0 ceil(max(b))],'Ylim',[min(amps)\*0.2 max(amps)\*1.1],'YColor','r')

set(h2,'Color','r')

%xlabel('time (minutes)')

xlabel('time (seconds)')

catch

text(0,0.5,'no timestamps','FontName','Courier')

axis off

end

In “mpsth2” function, we modified this part; pre=-100ms and post=900ms

In “mpsth11” function, we modified this part; pre=-100000ms (-100sec) and post=300000ms (300sec).

Download a function, Square\_coloring from here.

https://ossyaritoori.hatenablog.com/entry/2016/11/14/MATLAB\_figure%E3%81%AE%E8%83%8C%E6%99%AF%E3%82%92%E5%88%86%E3%81%91%E3%81%A6%E5%A1%97%E3%82%8A%E3%81%A4%E3%81%B6%E3%81%99\_%EF%BC%88%E8%A3%9C%E9%A1%8C%EF%BC%9A3%E6%AC%A1Spline%E8%A3%9C%E9%96%93

For **PAG** data, run “s2\_WaveClus\_to\_Mlib\_**PAG.m**” , then save the variable “trialspxHist{n,i}” (time code of the spike timing within -100 to +300 sec relative to blue light per the unit) with re-writing the name to such as “PAG\_Rat#\_depth(um)\_Cluster#\_Channel”. Those are stored in “ToMakePAG\_String.mat”. Next, run “Make\_PAG\_String.m” which saves the data ID in {:,1}, the spike time code in {:,2}, and the 1 sec binning histogram data in {:,3} of “PAG\_String.mat”.

When you run “Fig4B\_PAG\_Raster.m”, save the variable “PAG\_percent\_activity” in “PAG\_percent\_activity.mat”(where normalized raster map is stored) and save “PAG\_Conv1\_400{i,9}” (where percentage activity is stored)” in “PAG\_Conv1\_400\_PAG\_String.mat”. Using this variable, run “Fig4D\_PAG\_MeanCurve.m”, “Fig4E\_PAG\_Bars.m”, “FigS4D\_S4E\_PAG\_Latency\_ActiveCells.m”.

For **Spinal Cord** data, run “s2\_WaveClus\_to\_Mlib\_**SpinalCord.m**”, then save “trialspx1{k,1}{i,1}” (time code of the spike timing within -100 to +900 sec relative to the electric shock) with re-writing the name to such as “Rat#RscClus#”. Save also “ES2stIND(k,1)” (the time code of final ES during the first 290sec recording) with re-writing the name to such as “Rat#RscES2stIND”. Those are stored in “ToMakeWT (or ChR2 or dOVTChR2) String.mat”. Next, run “MakeWT (or ChR2 or dOVTChR2) String.m”. It saves the data ID in {:,1}, the raster time code for the entire electric shock period in {:,2}, the time code of final electric shock during the first 290sec recording in {:,3}, the raster time code for the 1st 290sec period in {:,4}, and the raster time code for the 2nd 60sec period in {:,5} of “WT (or ChR2 or dOVTChR2) String” in “SpinalCord\_String.mat”.

Then you can run “Fig5BC\_SC\_MeanCurve\_Bars\_FigS5B\_Kinetics.m”.

Run “SC\_MeanRasterPreparation.m” and save “WT(or ChR2 or dOVT)Conv290(or 60)\_1000” in “SpinalCord\_Conv.mat”. Using those variables, run “Fig5D\_SC\_MeanRaster\_XXX.m”.