**SleepAnal2.**

close all

clear all

prompt = {'Enter path to folder: ', ...

 'Enter input data .xlsx file name: ', ...

 'Enter .xlsx data range: ', ...

 'Enter the Group and Filter .xlsx file name: ',...

 'Enter the Group range: ', ...

 'Enter the Filter range: ', ...

 'Enter the zero date and time of your data (A1:B1 in your excel file). Use the format dd/mm/yyyy HH:MM: ', ...

 'Enter the start date and time for analysis. Use the format dd/mm/yyyy HH:MM: ', ...

 'Enter the end date and time for analysis. Use the format dd/mm/yyyy HH:MM: ', ...

 'Enter window in minutes [30]:', ...

 'Enter minimal sleep bout in minutes [5]:', ...

 'Plot standard errors (Y/N [N])? ', ...

 'Enter output .xlsx file name: '...

 'Plot individual Plots? (Y/N [N])?'};

 TNow = datestr(now, 'dd\_mm\_yy');

 DefAns = {'', '', '', '', '', '', '', '', '', '30', '5', 'N', ['Sleep' TNow '.xlsx'], 'N'};

 opts=struct('resize','on','windowstyle','normal','interpreter','none');

 Input = inputdlg(prompt,'Sleep Analysis', [1 50], DefAns, opts);

 %read data table

 Folder = Input{1};

 cd(Folder);

FileName = Input{2};

Range = Input{3};

f = waitbar(0.5,'Please wait, reading data...');

DataTable = readtable(FileName, 'range', Range);

close(f)

%read grouping and filter table

GFFile = Input{4};

GRange = Input{5};

FRange = Input{6};

f = waitbar(0.5,'Please wait, reading filter data...');

Group = xlsread(GFFile, GRange);

Filter = xlsread(GFFile, FRange);

close(f)

Zdate = Input{7};

Sdate = Input{8};

Edate = Input{9};

Window = str2double(Input{10});

MinBout = str2double(Input{11});

if strcmp(Input{12}, 'Y')%Plot SE?

 PlotSE = 1;

else

 PlotSE = 0;

end

OutFileName = Input{13};%output file

if strcmp(Input{14}, 'Y')%Plot Individual plots?

 PlotIndiv = 1;

else

 PlotIndiv = 0;

end

GR = unique(Group(~Filter & Group ~= 0));

Colors = cell(1, length(GR));

for j = 1:length(GR)

 Colors{j} = uisetcolor([0 0 0], ['Select a color for Treatment ' num2str(GR(j))]);

end

f = waitbar(0.5,'Please wait... plotting');

[T, Sleep, SleepRaw, MnSlp, SEMSlp, x, P] = SleepPlotV2(DataTable, Group', Filter',...

 Zdate, Sdate, Edate, Window, MinBout, PlotSE, Colors);

legend([P{:}], 'orientation', 'horizontal', 'location', 'best');

legend('boxoff');

print('SleeMeans', '-dtiff', '-r300');

close(f)

%write to .xlsx file

f = waitbar(0.5,'Please wait... writing to xlsx file');

 for j=1:length(GR)

 writetable(SleepRaw{j}, OutFileName , 'Sheet', ['Treat ' num2str(GR(j))]);

 end

close(f)

%individual sleep plots

if PlotIndiv ==1

 FilterIdx = find(~Filter);

 s = sum(~Filter);

 if mod(s, 3) >= 1

 NumFigs = floor(s/3)+1;

 else

 NumFigs = floor(s/3);

 end

 c = 2;

 for j = 1:3:s

 figure(c)

 for l = 1:3

 subplot(4, 1, l)

 SleepPlotV2(DataTable(:, FilterIdx(j+l-1)), 1, 0, Zdate, Sdate, Edate, Window, MinBout, 0, {'black'});

 title(DataTable(:, FilterIdx(j+l-1)).Properties.VariableNames{:});

 hold off

 end

 c = c+1;

 PaperSet\_portrait;

 set(gcf, PaperSet{:})

 print(['Individuals\_' num2str(c)], '-dtiff', '-r300');

 end

end

% bouts analysis

**SleepPlotV2.m**

function [T, Sleep, SleepRaw, MnSlp, SEMSlp, x, P] = SleepPlotV2(Table, Group, Filter, ZDate, StartDate, EndDate, Window, MinBout, PlotSE, Colors)

% Days- nuumber of days to analyze from beginning of recording

% Window- sliding window in minutes for precent sleep caculation

%start and end indices

ZNum = datenum(ZDate, 'dd/mm/yyyy HH:MM');

StartNum = datenum(StartDate, 'dd/mm/yyyy HH:MM');

EndNum = datenum(EndDate, 'dd/mm/yyyy HH:MM');

StartIdx = floor((StartNum - ZNum) \* 24\*60)+1;

EndIdx = floor((EndNum - ZNum) \* 24\*60)+1;

% make time table

T = Table(StartIdx:EndIdx, :);

TotalTime = ZNum : 1/24/60: ZNum + height(Table)\*1/24/60; % total time vector

TimeNum = TotalTime(StartIdx : EndIdx); % specified time vector

TimeAnal = datetime(datevec(TimeNum), 'Format', 'dd/MM/yyyy HH:mm');

GR = unique(Group(~Filter & Group ~= 0));

Sleep = cell(1, length(GR));

SleepRaw = cell(1, length(GR));

c = 1;

for g = GR

 Sleep{c} = zeros(height(T), sum(Group == g & ~ Filter));

 ZeroAct = T{:, Group == g & ~ Filter} ~= 0; % zero activity logical vector

 for h = 1:size(ZeroAct,2)

 temp = zeros(size(ZeroAct, 1), 1); % sleep logical vector

 for j = 1:size(ZeroAct, 1) - MinBout - 1

 if sum(ZeroAct(j:j + MinBout - 1, h)) == 0

 k = j;

 while ZeroAct(k, h) == 0

 temp(k) = 1;

 k = k + 1;

 if k == size(ZeroAct, 1)

 break

 end

 end

 j = k;

 end

 end

 SleepRaw{c}(:, h) = temp;

 temp = [zeros(Window, 1); temp];

 SleepSlide = zeros(length(temp), 1);

 for j = Window : length(temp)

 SleepSlide(j) = 100 \* sum(temp(j - (Window - 1) : j)) / Window;

 end

 Sleep{c}(:, h) = SleepSlide(Window + 1 : end);

 end

 Sleep{c} = array2table(Sleep{c}, 'variablenames',...

 T(:, Group == g & ~ Filter).Properties.VariableNames);

 SleepRaw{c} = array2table(SleepRaw{c}, 'variablenames',...

 T(:, Group == g & ~ Filter).Properties.VariableNames);

 SleepRaw{c} = [table(TimeAnal, 'VariableNames', {'Time'}), SleepRaw{c}];

 MnSlp{c} = mean(Sleep{c}{:,:}, 2);

 MnSlp{c} = MnSlp{c}(Window : end, :);

 SEMSlp{c} = std(Sleep{c}{:,:}, [], 2)./sqrt(size(Sleep{c}, 2)-1);

 SEMSlp{c} = SEMSlp{c}(Window : end, :);

 c = c+1;

end

PaperSet\_landscape;

set(gcf, PaperSet{:});

x= TimeNum(Window : end);

for g = 1: size(GR, 2)

 if PlotSE

 fill([x'; flipud(x')], [MnSlp{g} - SEMSlp{g}; flipud(MnSlp{g} + SEMSlp{g})], Colors{g}, 'facealpha', 0.2, 'linestyle', 'none');

 end

 hold on

 P{g} = plot(x, MnSlp{g}', 'color', Colors{g}, 'linewidth', 0.5);

end

xtick = floor(StartNum) : 0.5 : ceil(EndNum);

xticks(xtick);

%xlim([xtick(1) xtick(end)]);

xticklabels(datestr(xtick, 'HH:MM'));

set(gca, 'box', 'off', 'tickdir', 'out');

set(gcf, PaperSet{:});

end

PaperSet\_portrait.m

PaperSet = {'paperunits', 'centimeters', 'papertype', 'a4', 'paperposition', ...

 [0.5 0.5 19.05 27.51], 'paperorientation', 'portrait'};

 set(gcf, PaperSet{:});