```
%% By KEITH VOSSEL Lab, David Geffen School of Medicine, University of California, Los Angeles.
%
clear;
disp(datetime)

% edf files location
filePattern_edf = fullfile('C:\Files\**', '*.edf'); % All visits\
matFiles_edf = dir(filePattern_edf);

% struct files location, previously saved by Delphos script
filePattern_20 = fullfile('C:\Files\**', '*.mat');  % All visits\
matFiles_20 = dir(filePattern_20);

% location to save any files
folder_output = 'C:\Files\';

labels_ripples = zeros(272,length(matFiles_20));
labels_fast = zeros(272,length(matFiles_20));

for m =  1 : length(matFiles_edf)
    if contains(matFiles_edf(m,1).name,"meg")

        filename_in = append(matFiles_edf(m).folder, '\',matFiles_edf(m).name);
        header_in = ft_read_header(filename_in);
        Fs = header_in.Fs;
        data_in = ft_read_data(filename_in, 'header',header_in);

        file_20 = append(matFiles_20(m).folder, '\',matFiles_20(m).name);
        results_20 = load(file_20);
        results_20 = results_20.results_20;

        % to check sample rate for each file
        if Fs > 1201
            samples = size(data_in);

            if Fs == 4000
                length_4000_12000 = samples(2) * 3;
                data_Fs_12000 = zeros(samples(1), length_4000_12000);
                for i = 1 : samples(2)
                    for j = 1 : samples(1)
                        data_Fs_12000(j, 3*i-2 : 3*i) = data_in(j, i);
                    end
                end
                length_12000_1200 = length_4000_12000 / 10;
                data_Fs = zeros(samples(1), length_12000_1200);
                for i = 1 : length_12000_1200
                    for j = 1 : samples(1)
                        data_Fs(j, i) = mean(data_Fs_12000(j, 10*i-9 : 10*i));
                    end
```

```
            end
            data_in = data_Fs;
        end

        if Fs == 2400
            length_2400_1200 = samples(2)/2 ;
            data_Fs = zeros(samples(1), length_2400_1200);
            for i = 1 : length_2400_1200
                data_Fs(:, i) = (data_in(:, (2*i - 1)) + data_in(:, (2*i))) / 2 ;
            end
            data_in = data_Fs;
        end
    else
    end

    %% highpass filter the signal and check all hfo candidate's standard deviation

    disp(['in filter of ',num2str(m)])
    data_hpf1 = zeros(size(data_in));
    nChans = header_in.nChans;
    fpass1 = 5;
    Wn1 = fpass1 / (Fs/2); % Normalized cutoff frequency
    [b, a] = butter(4, Wn1, 'high'); % Create filter coefficients

    for i = 1:nChans
        data_hpf1(i,:) = filtfilt(b, a, data_in(i,:)); % Apply filter to signal
    end

    data_in_1000 = 1000 * data_in;
    nChans = header_in.nChans;
    data_hpf80 = zeros(size(data_in_1000));
    fpass = 80;
    Wn = fpass / (Fs/2); % Normalized cutoff frequency
    [b, a] = butter(4, Wn, 'high'); % Create filter coefficients

    for i = 1:nChans
        data_hpf80(i,:) = filtfilt(b, a, data_in_1000(i,:)); % Apply filter to↙
signal
    end

    stdev_thres = 3;
    bin_size = Fs * 0.4;
    samples = size(data_hpf80);
    data_hpf80_std_3 = zeros(samples(1), samples(2));
    data_hpf80_std_2 = zeros(samples(1), samples(2));
    bins = samples(2) / bin_size;

    for i = 1 : nChans

        for j = 1 : bins
```

```
            start_sample = bin_size*(j-1) + 1;
            end_sample = bin_size*j;
            stdev = std(data_hpf80(i, start_sample : end_sample));

            for k = 1 : bin_size
                current_sample = start_sample + (k-1);
                ratio = data_hpf80(i, current_sample) / stdev;
                if ratio >= stdev_thres || ratio <= -stdev_thres
                    data_hpf80_std_3(i, current_sample) = 1 ;
                end
                if ratio >= (stdev_thres - 1) || ratio <= -(stdev_thres - 1)
                    data_hpf80_std_2(i, current_sample) = 1 ;
                end
            end
        end

        for j = 1 : (bins - 1)
            start_sample = bin_size*(j-1/2) + 1;
            end_sample = bin_size*(j+1/2);
            stdev = std(data_hpf80(i, start_sample : end_sample));

            for k = 1 : bin_size
                current_sample = start_sample + (k-1);
                ratio = data_hpf80(i, current_sample) / stdev;
                if ratio >= stdev_thres || ratio <= -stdev_thres
                    data_hpf80_std_3(i, current_sample) = 1 ;
                end
                if ratio >= (stdev_thres - 1) || ratio <= -(stdev_thres - 1)
                    data_hpf80_std_2(i, current_sample) = 1 ;
                end
            end
        end
    end
    % end of filter

    %% for list

    hfo_count_20 = length(results_20.results.markers);
    list_hfo_20 = [];
    list_other_20 = {};

    for i = 1:hfo_count_20
        list_hfo_20(i,1) = results_20.results.markers(1,i).position;
        list_hfo_20(i,3) = results_20.results.markers(1,i).power;
    end

    for i = 1:hfo_count_20
        list_other_20{i,1} = cell2mat(results_20.results.markers(1,i).channels);
    end
```

```
        for i = 1:hfo_count_20
            list_other_20{i,3} = double(list_other_20{i,1});
            sum_char = list_other_20{i,3}(2)*1e6 + list_other_20{i,3}(3)*1e4 + ↙
list_other_20{i,3}(4)*1e2 + list_other_20{i,3}(5)*1;
            list_other_20{i,2} = sum_char;
            list_hfo_20(i,2) = list_other_20{i,2};
        end

        list_hfo_20(:,35) = 0;   % initializing, till some column


        %% info matrices

        hfo_len_20 = min(hfo_count_20, length(list_hfo_20));
        mtx_1_20 = zeros(hfo_len_20);  mtx_2_20 = zeros(hfo_len_20);  mtx_3_20 = zeros ↙
(hfo_len_20);  mtx_4_20 = zeros(hfo_len_20);   % in case no mtx

        for i = 1:hfo_len_20
            for j = 1:hfo_len_20
                diff = list_hfo_20(i,1) - list_hfo_20(j,1);
                mtx_1_20(i,j) = diff^2;
                mtx_1_20(i,j) = mtx_1_20(i,j)^0.5;
            end
        end

        for i = 1:hfo_len_20
            for j = 1:hfo_len_20
                if mtx_1_20(i,j) < 0.1 && list_hfo_20(i,2) == list_hfo_20(j,2)
                    mtx_2_20(i,j) = 1 ;
                else
                    mtx_2_20(i,j) = 0 ;
                end
            end
        end

        for i = 1:hfo_len_20
            for j = 1:hfo_len_20
                if mtx_1_20(i,j) < 0.04 && list_hfo_20(i,2) ~= list_hfo_20(j,2)
                    if results_20.results.markers(1,i).power > results_20.results. ↙
markers(1,j).power
                        mtx_3_20(i,j) = 100000;
                    else
                        mtx_3_20(i,j) = 10000000;
                    end
                end
            end
        end

        for i = 1:hfo_len_20
            for j = 1:hfo_len_20
```

```
                if mtx_1_20(i,j) < 0.2 && list_hfo_20(i,2) == list_hfo_20(j,2) && i~=j
                    mtx_4_20(i,j) = 1000 ;
                end
            end
        end

        for i = 1:hfo_len_20
            list_hfo_20(i,4) = sum(mtx_2_20(i,:));
            list_hfo_20(i,5) = sum(mtx_3_20(i,:));
            list_hfo_20(i,6) = sum(mtx_4_20(i,:));
        end

        list_hfo_20(:,7) = list_hfo_20(:,4) + list_hfo_20(:,5) + list_hfo_20(:,6) ;

        %% identify qualified ones

        numK = 000;
        numO = 4;
        numM = 100000;
        KOnum = (numK + numO);

        for k = 1:hfo_len_20
            if results_20.results.markers(1,k).value >= 80  ...
                    && results_20.results.markers(1,k).value < 500  ...
                    && string(results_20.results.markers(1,k).label) ~=  "Spike"
                if        (list_hfo_20(k,7) <= KOnum)  ...
                        || ((list_hfo_20(k,7) >= ( 1 * numM + 0 * KOnum)) && ↙
(list_hfo_20(k,7) <= ( 1 * numM + 1 * KOnum)))  ...
                        || ((list_hfo_20(k,7) >= ( 2 * numM + 0 * KOnum)) && ↙
(list_hfo_20(k,7) <= ( 2 * numM + 1 * KOnum)))  ...
                        || ((list_hfo_20(k,7) >= ( 3 * numM + 0 * KOnum)) && ↙
(list_hfo_20(k,7) <= ( 3 * numM + 1 * KOnum)))  ...
                        || ((list_hfo_20(k,7) >= ( 4 * numM + 0 * KOnum)) && ↙
(list_hfo_20(k,7) <= ( 4 * numM + 1 * KOnum)))  ...
                        || ((list_hfo_20(k,7) >= ( 5 * numM + 0 * KOnum)) && ↙
(list_hfo_20(k,7) <= ( 5 * numM + 1 * KOnum)))
                    list_hfo_20(k,12) = 1;
                end
            end
        end

        %% check if more/less than stdev-threhold from unfiltered data

        count_labels = length(results_20.results.labels);
        label_mtx = zeros(count_labels, 1);

        for i_label = 1 : count_labels
            label_no = double(cell2mat(results_20.results.labels(i_label,1)));
            sum_char = label_no(2)*1e6 + label_no(3)*1e4 + label_no(4)*1e2 + label_no ↙
(5)*1 ;
```

```
            label_mtx(i_label,1) = sum_char;
        end

        no_of_std_thres = 1;

        cycles = 4;
        for i_list_hfo_20 = 1 : hfo_count_20
            if list_hfo_20(i_list_hfo_20, 12) == 1
                for i_label_mtx = 1 : count_labels
                    if list_hfo_20(i_list_hfo_20, 2) == label_mtx(i_label_mtx, 1)
                        hfo_sample_position_range = cycles / (results_20.results.
markers(1, i_list_hfo_20).value);
                        hfo_sample_position_start = round( ( list_hfo_20(i_list_hfo_20,
1) - hfo_sample_position_range ) * Fs );
                        hfo_sample_position_end = round( ( list_hfo_20(i_list_hfo_20,
1) + hfo_sample_position_range ) * Fs );
                        if hfo_sample_position_end < samples(2)
                            sum_square_hfo_std_3 = sumsqr( data_hpf80_std_3
(i_label_mtx, hfo_sample_position_start : hfo_sample_position_end) ) ;
                            sum_square_hfo_std_2 = sumsqr( data_hpf80_std_2
(i_label_mtx, hfo_sample_position_start : hfo_sample_position_end) ) ;
                            if (sum_square_hfo_std_3 > (1 * no_of_std_thres)) &&
(sum_square_hfo_std_2 > (5 * no_of_std_thres))
                                list_hfo_20(i_list_hfo_20, 32) = 1;
                            end
                        end
                    end
                end
            end
        end

        %% count hfos for each channel
        for i_c = 1 : hfo_len_20
            if list_hfo_20(i_c, 32) == 1
                for p = 1 : length(results_20.results.labels)
                    if string(results_20.results.markers(1,i_c).channels) == string
(results_20.results.labels{p,1})
                        if (results_20.results.markers(1,i_c).value) <= 250
                            labels_ripples(p,m) = labels_ripples(p,m) + 1;
                        elseif (results_20.results.markers(1,i_c).value) > 250
                            labels_fast(p,m) = labels_fast(p,m) + 1;
                        end
                    end
                end
            end
        end
        disp(['over results of ',num2str(m)])
    end
end
```

```
% for creating excel
writematrix(labels_ripples', [folder_output , 'labels_ripples_' , folder_output(end-3:end-1) , '.xlsx']);
writematrix(labels_fast', [folder_output , 'labels_fast_' , folder_output(end-3:end-1) , '.xlsx']);
disp(datetime)
```