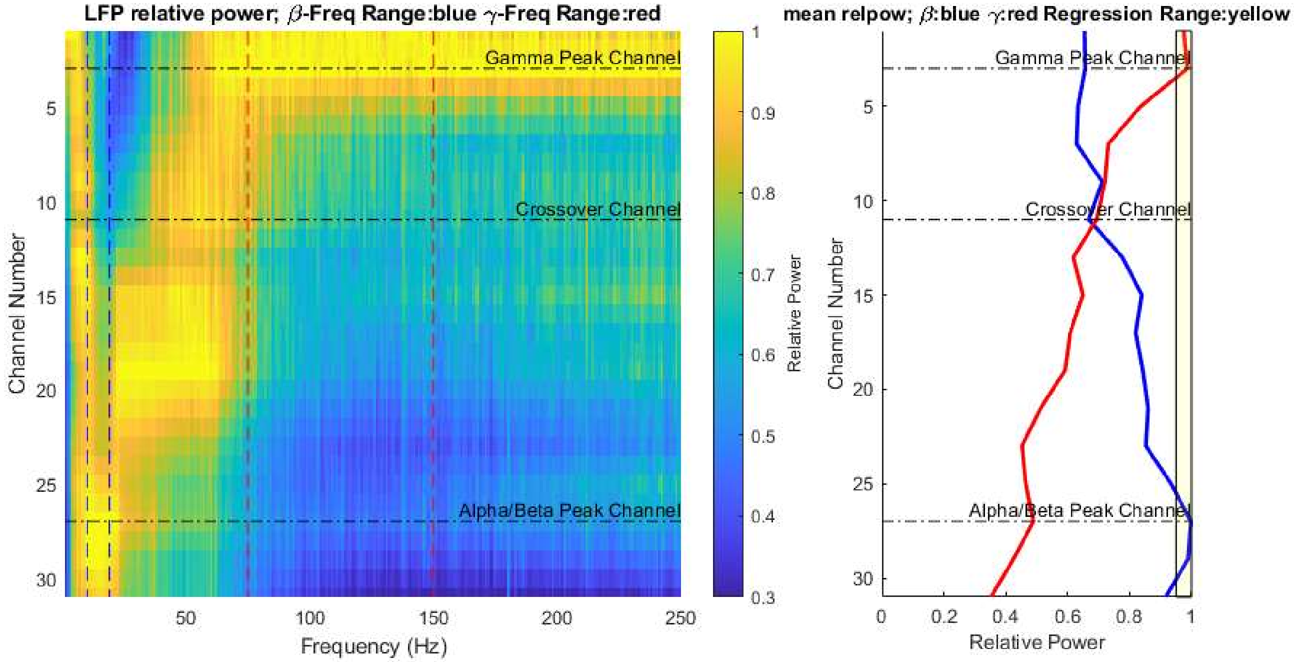
```
% Follow along this script with the instructions: "FLIP Algorithm Introduction"

path = '\\millerdata.mit.edu\amajor\Earl Miller Lab\MANUSCRIPTS\Spectrolaminar pattern\Minor Revisions_10192023\2023-11-02 code_AM\';
file = 'data.mat';

% disp('Select the data.mat file from the directory in which it is saved')
% pause(1);
% [file, path] = uigetfile;
load([path file])


%%%%%%%%%%%% Example 1: FLIP Using Default Values
probe1 = squeeze(data.relpow(863, :, :));
probe1(any(isnan(probe1), 2), : ) = []; % remove NaN rows
disp(['power map is size: ' num2str(size(probe1))]);
laminaraxis = 0:0.1:3;
freqaxis = 1:250;
setfreqbool = 1;

% run FLIP with default frequency bands
[startinglowfreq,endinglowfreq,startinghighfreq,endinghighfreq,goodnessvalue,superficialchannel,deepchannel,highfreqmaxchannel,lowfreqmaxcha
nnel,crossoverchannel] = ...
    FLIPAnalysis(probe1,laminaraxis,freqaxis,setfreqbool);


%%%%%%%%%%%% Example 2: FLIP Using User-Defined Frequency Bin Values
lowfreqrange = [1 30];
highfreqrange = [100 140];
[startinglowfreq,endinglowfreq,startinghighfreq,endinghighfreq,goodnessvalue,superficialchannel,deepchannel,highfreqmaxchannel,lowfreqmaxcha
nnel,crossoverchannel] = ...
    FLIPAnalysis(probe1,laminaraxis,freqaxis,setfreqbool, lowfreqrange, highfreqrange);


%%%%%%%%%%%% Example 3: Finding Optimal Frequency Bin Values with vFLIP
probe2 = squeeze(data.relpow(807, :, :));
probe2(any(isnan(probe2), 2), : ) = [];
setfreqbool = 0;
[startinglowfreq,endinglowfreq,startinghighfreq,endinghighfreq,goodnessvalue,superficialchannel,deepchannel,highfreqmaxchannel,lowfreqmaxcha
nnel,crossoverchannel] = ...
    FLIPAnalysis(probe2,laminaraxis,freqaxis,setfreqbool);


%%%%%%%%%%%% Example 4: Generating the spectrolaminar pattern (a.k.a. relative power map) from example LFP data #1
% NOTE: fieldtrip package is required for code beyond this point
% https://www.fieldtriptoolbox.org/download/
% https://www.fieldtriptoolbox.org/faq/should_i_add_fieldtrip_with_all_subdirectories_to_my_matlab_path/
% addpath('/your/directory/for/fieldtrip-20231025');
addpath('\\millerdata.mit.edu\common\Alex\fieldtrip-20191111;');
ft_defaults;
global ft_default
ft_default.showcallinfo = 'no';
ft_default.trackusage = 'no';
ft_warning off;

lfp1 = data.example1_vlPFC_lfp;
[relpow1] = relpow_from_rawLFP(lfp1); % lfp should be in dimension format: nchans x trialtime x ntrials % assumes lfp data in 1 kHz resoluti
on % requires fieldtrip toolbox
FLIPAnalysis(relpow1,0:size(relpow1,1)-1,1:size(relpow1,2),1); % plot


%%%%%%%%%%%% Example 5: Generating the spectrolaminar pattern (a.k.a. relative power map) from example LFP data #2
lfp2 = data.example2_7A_lfp;
[relpow2] = relpow_from_rawLFP(lfp2);
FLIPAnalysis(relpow2,0:size(relpow2,1)-1,1:size(relpow2,2),1); % plot


%%%%%%%%%%%% Example 6: Plotting Sample Normalized Power Data from any probe in relpow matrix
row = 752; % there are 942 rows (942 probes in data.relpow) % pick any row
relpow_any_row = squeeze(data.relpow(row,:,:));
relpow_any_row(any(isnan(relpow_any_row), 2), : ) = [];
FLIPAnalysis(relpow_any_row,0:size(relpow_any_row,1)-1,1:size(relpow_any_row,2),1); % plot


%%%%%%%%%%%% Example 7: Generate Area-average Spectrolaminar Patterns
brain_area = 'MST';
brain_area_num = 2;
index = find([data.meta.brain_area_num]' == brain_area_num);
mean_relpow = squeeze(nanmean(data.relpow(index,:,:),1));
FLIPAnalysis(mean_relpow,0:size(mean_relpow,1)-1,1:size(mean_relpow,2),1); % plot
sgtitle(['mean LFP relative power (n = ' num2str(length(index)) ')']);

%%%%%%%%%%%% Example 8: Plot Current Source Density for any probe
```

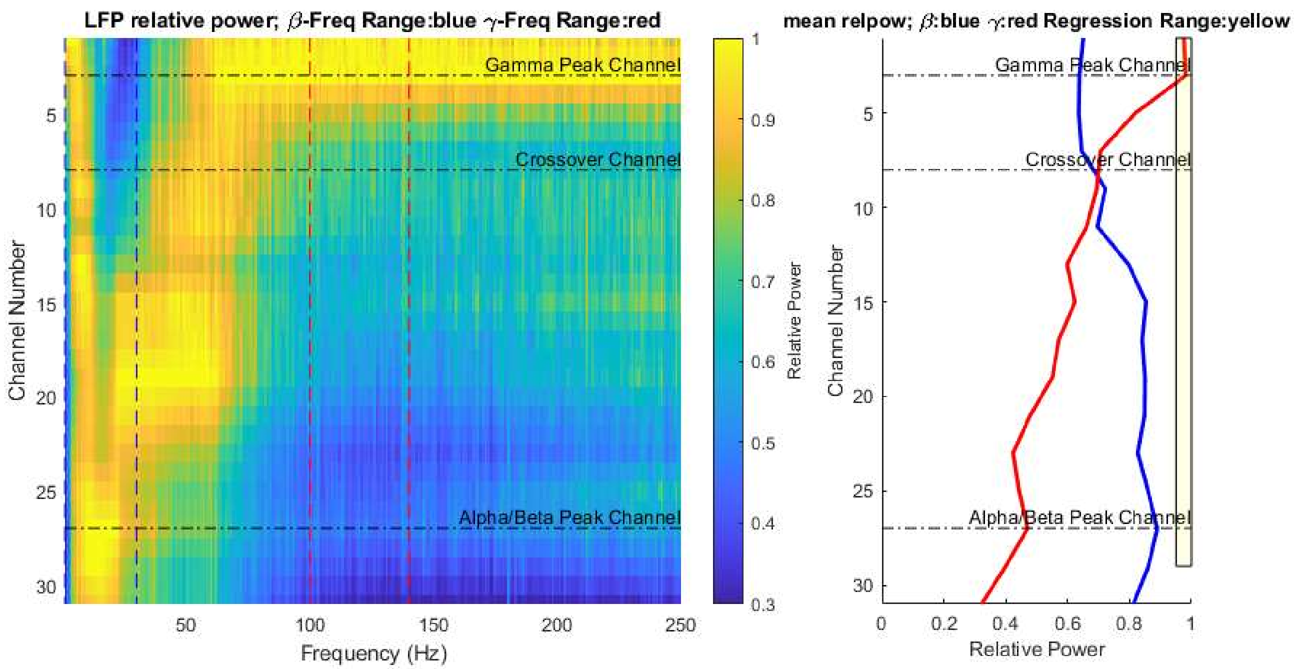
```
row = 752;
CSD = squeeze(data.CSD(row,:,:));
CSD(all(isnan(CSD),2),:) = []; % remove NaN rows
x = figure; imagesc(CSD);set(gca, 'YDir', 'reverse');
ylim([1 size(CSD,1)]); title('CSD'); xlabel('Time (ms)'); ylabel('Channel');
cb=colorbar; ylabel(cb, 'Normalized CSD');




function [relpow] = relpow_from_rawLFP(lfp)

% lfp should be in format nchans x trialtime x ntrials
% assumes lfp data in 1 kHz resolution
% require fieldtrip toolbox

foi = 1:150; % frequencies of interest

nchannels = size(lfp, 1); ntrials = size(lfp, 3); ntrialtime =  size(lfp, 2);
% prep ft format
dataLFP = [];
for t = 1:ntrials
    dataLFP.time{t,1} = 0.001:0.001:ntrialtime*0.001; % in seconds
    dataLFP.trial{t,1} = zeros(nchannels, ntrialtime); %nchannels x ntrialtime
    dataLFP.trial{t,1} = lfp(:, :, t);

end
dataLFP.fsample = 1000;
dataLFP.label = {};
%make each channel unique
for c = 1:nchannels
    dataLFP.label{c} = ['ch' num2str(c)];
end
dataLFP.trialinfo = zeros(ntrials, 1);
dataLFP.trialinfo(:, 1) = 1:ntrials;


cfg = [];
cfg.method = 'mtmfft';
cfg.taper = 'hanning';
cfg.output = 'pow';
cfg.keeptrials = 'yes';
cfg.foi = foi;
cfg.pad = 'nextpow2';
pow = ft_freqanalysis(cfg, dataLFP);
meanpow = squeeze(mean(pow.powspctrm));
relpow = meanpow ./ max(meanpow);
end
```

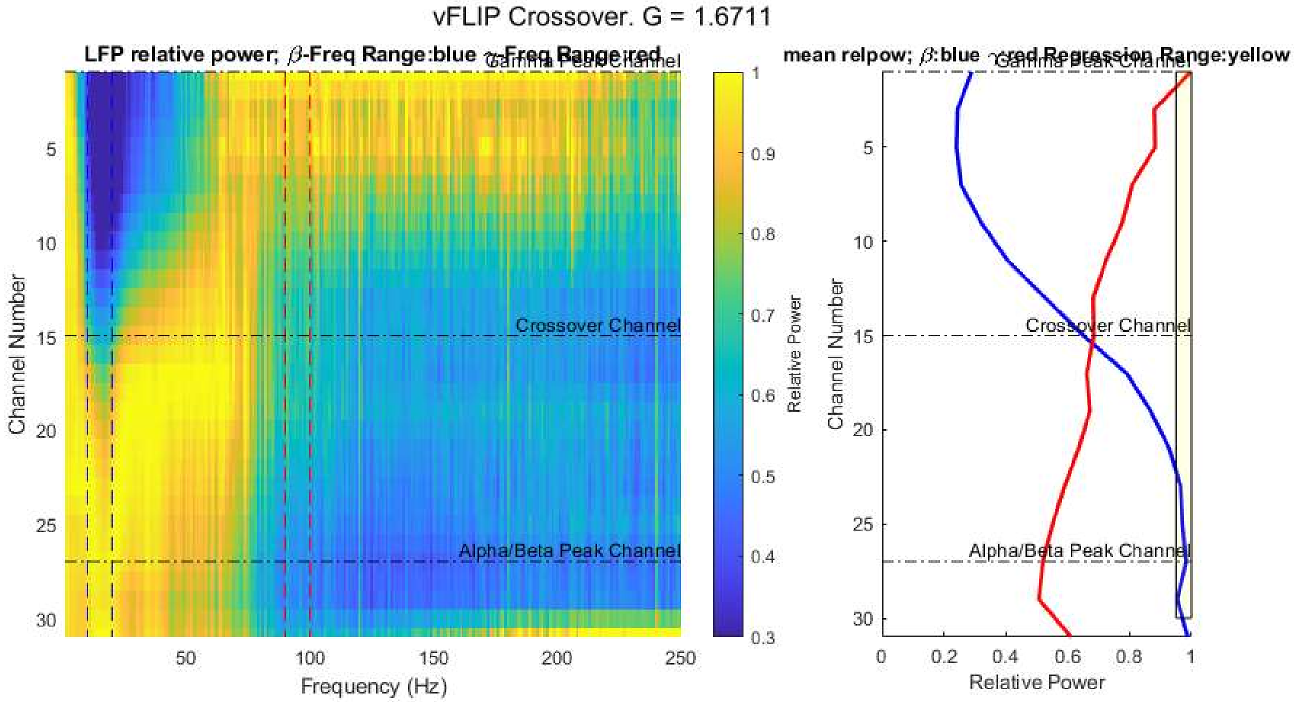
```
power map is size: 31  250
```

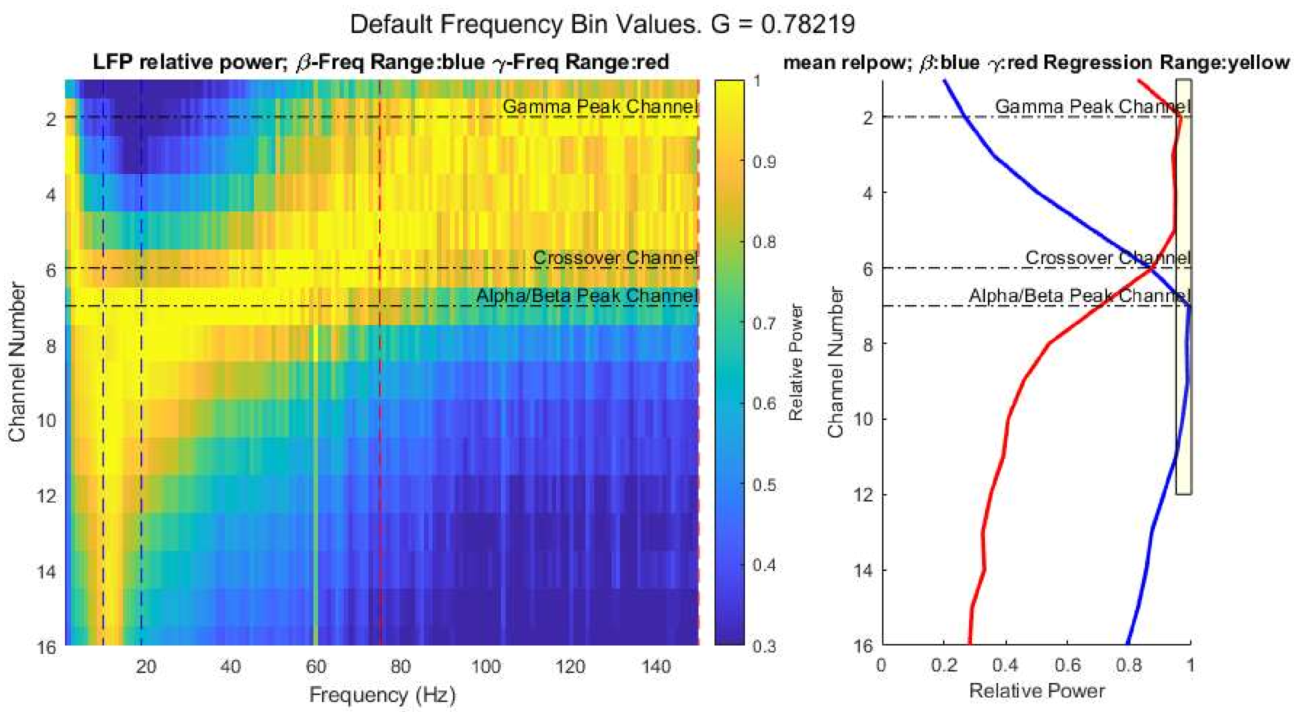
Default Frequency Bin Values. G = 1.6792

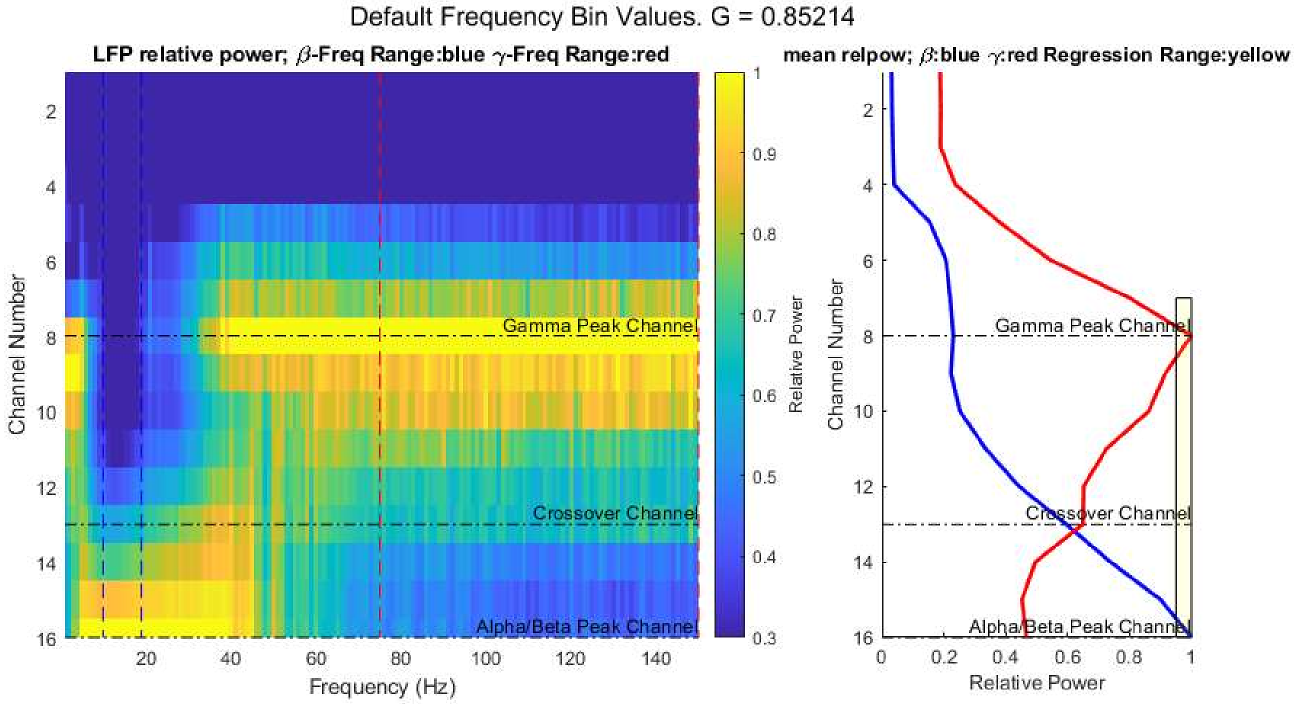


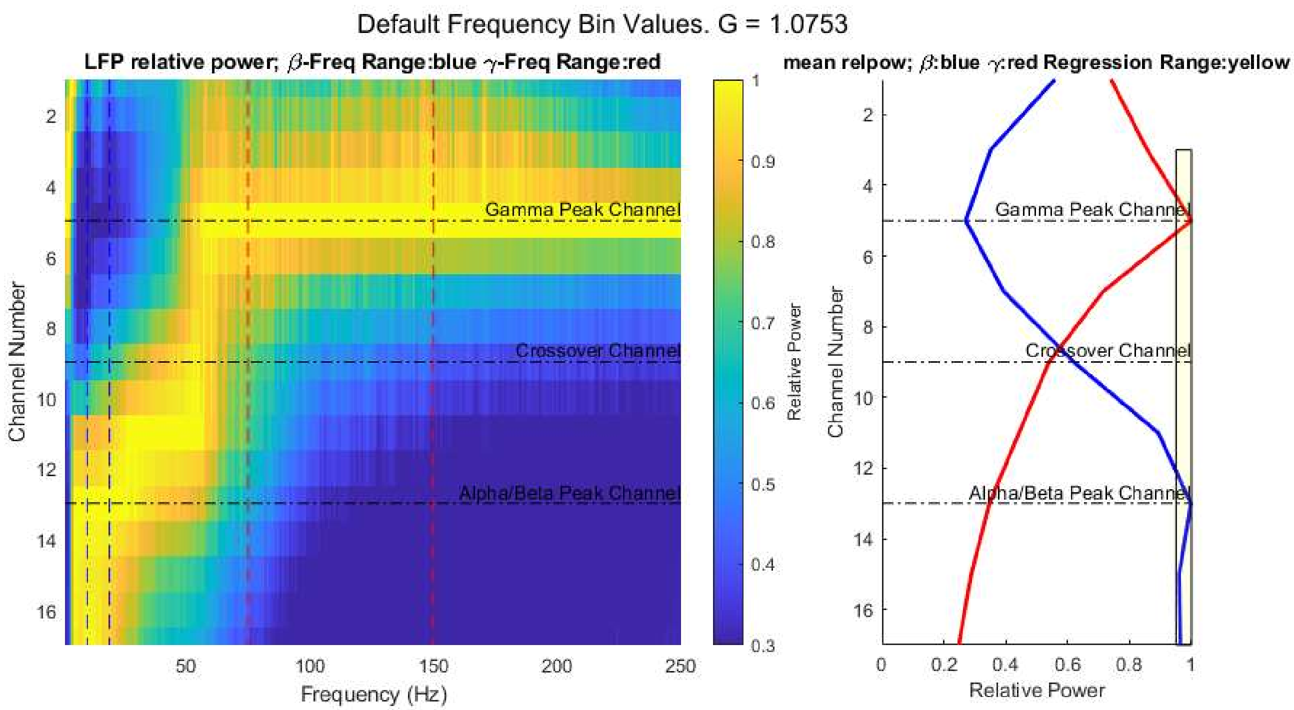
Set Frequency Bin Values. G = 1.4549

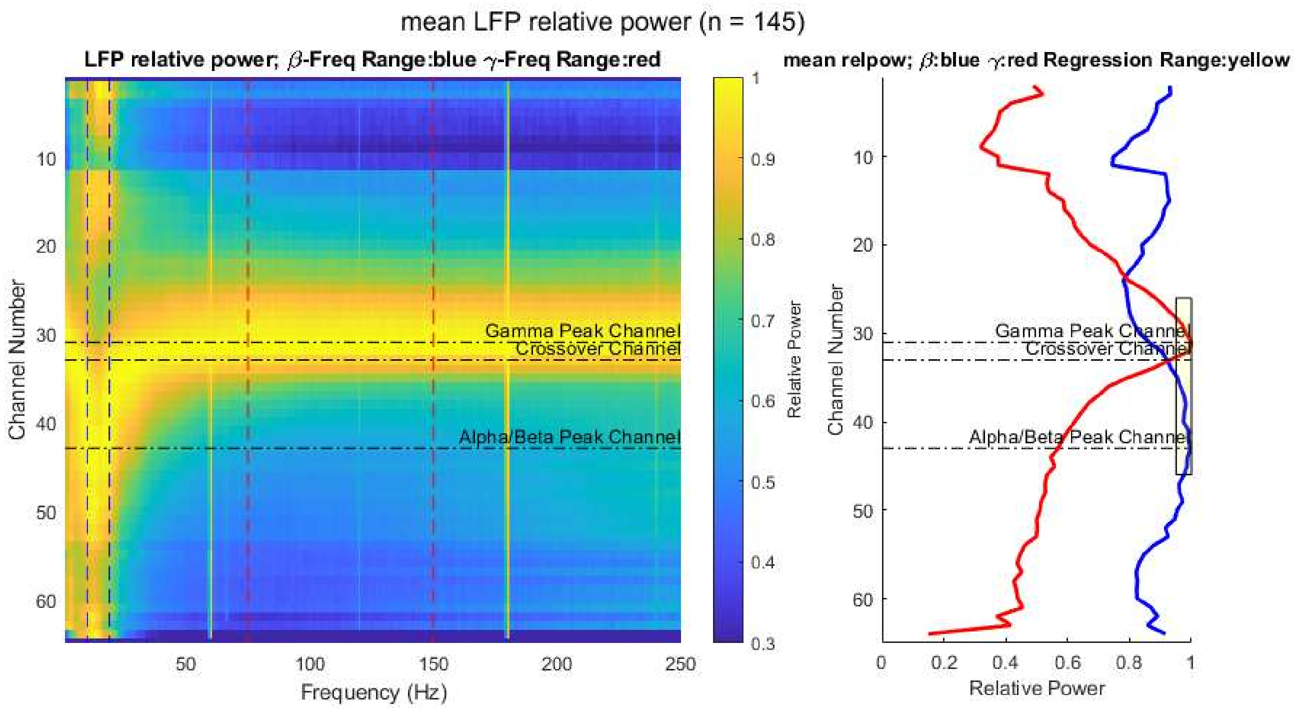


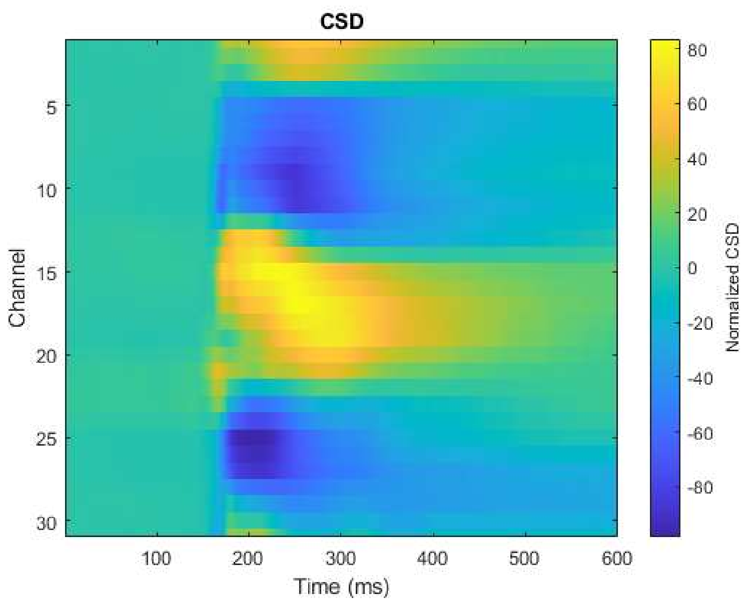

vFLIP Crossover. G = 1.6711
LFP relative power; β-Freq Range:blue γ-Freq Range:red
Gamma Peak Channel
Crossover Channel
Alpha/Beta Peak Channel
Channel Number
Frequency (Hz)
Relative Power
mean relpow; β:blue γ:red Regression Range:yellow


Default Frequency Bin Values. G = 0.78219
LFP relative power; β-Freq Range:blue γ-Freq Range:red
Gamma Peak Channel
Crossover Channel
Alpha/Beta Peak Channel
Channel Number
Frequency (Hz)
Relative Power
mean relpow; β:blue γ:red Regression Range:yellow


Default Frequency Bin Values. G = 0.85214
LFP relative power; β-Freq Range:blue γ-Freq Range:red
mean relpow; β:blue γ:red Regression Range:yellow
Gamma Peak Channel
Crossover Channel
Alpha/Beta Peak Channel
Channel Number
Frequency (Hz)
Relative Power


Default Frequency Bin Values. G = 1.0753
LFP relative power; β-Freq Range:blue γ-Freq Range:red
mean relpow; β:blue γ:red Regression Range:yellow
Gamma Peak Channel
Crossover Channel
Alpha/Beta Peak Channel
Channel Number
Frequency (Hz)
Relative Power


mean LFP relative power (n = 145)
LFP relative power; β-Freq Range:blue γ-Freq Range:red
Channel Number
Frequency (Hz)
Relative Power
Gamma Peak Channel
Crossover Channel
Alpha/Beta Peak Channel
mean relpow; β:blue γ:red Regression Range:yellow
Channel Number
Relative Power
Gamma Peak Channel
Crossover Channel
Alpha/Beta Peak Channel


CSD
Channel
Time (ms)
Normalized CSD