I (Jacob Salminen) believe there to be several plausible ways to baseline event related spectral perturbations. Below are several approaches using software available through EEGLAB and more basic statistics.

# newtimefbaseln.m (EEGLAB)

## [outputs] = function(inputs):

[PP, baseln, mbase] = newtimefbaseln(PPori, timesout, varargin)

*PPori*: is a cell that typically contains time frequency data contained in a cell the data is of the format (FREQUENCIESxTIMESxITERATIONS) where iterations could be trials or subjects.

timesout: is a column vector (1xN) of the times you want to baseline in *PPori*.

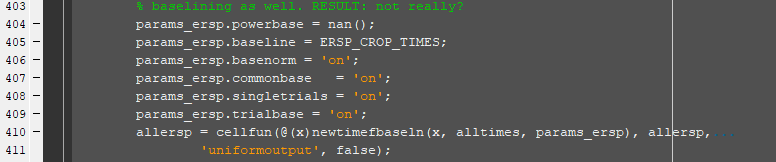
## options:

| **Parameter Name** | **Parameter Type** | **Parameter Options** | **Parameter Default** |
| --- | --- | --- | --- |
| 'powbase' | ‘real’ | [] | NaN |
| ‘basenorm’ | ‘string’ | {‘on’,’off’} | ‘off’ |
| ‘baseline’ | ‘real’ | [] | 0 |
| ‘commonbse’ | ‘string’ | {‘on’,’off’} | ‘off’ |
| 'singletrials' | ‘string’ | {‘on’,’off’} | ‘on’ |
| 'trialbase' | ‘string’ | {'on','off','full'} | ‘off’ |
| 'verbose' | ‘string’ | {‘on’,’off’} | ‘on’ |

## options notes

* ‘powbase’: this allows a user to input their own averaged baseline values. Otherwise newtimefbaseln.m will calculate the mean for each cell in *PPori* depending on ‘singletrials’ & ‘trialbase’ parameters.
* ‘basenorm’: if ‘on’, normalizes baseline before removing it from *PPori*
* ‘baseline’: is an Nx2 array of time values in (ms).
* ‘commonbase’: if ‘on’, computes an averaged baseline for each baseline calculated for each cell in *PP*ori
* ‘singletrials’: if ‘on’ && ‘trialbase’ is ‘off’, will average across across ‘trials’ before calculating average across time
* ‘trialbase’: if ‘on’, this does some weird division stuff with the mean baseline calculated across trials & time.
* ‘verbose’: if ‘on’, does provide output prints.

## Testing

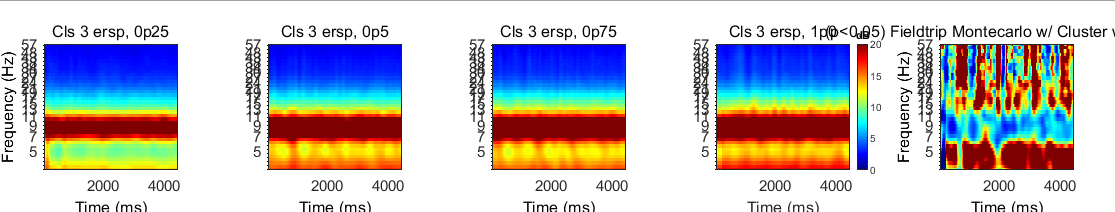


NOTE: Loaded data using std\_readdata() and loaded all trials (gait cycles) for a particular condition and subject.

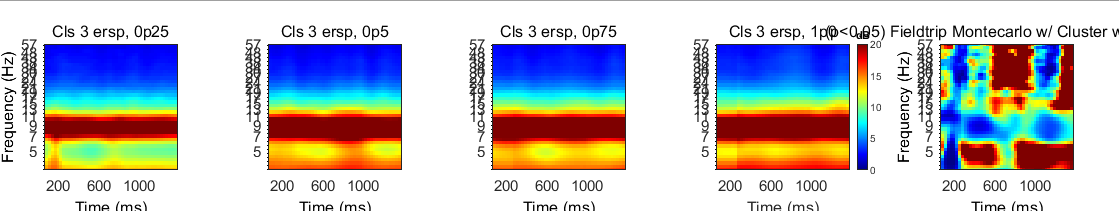
**Format::{‘powbase’,’basenorm’,’baseline’,’commonbase’,’singletrials’,’trialbase’,’verbose’}**

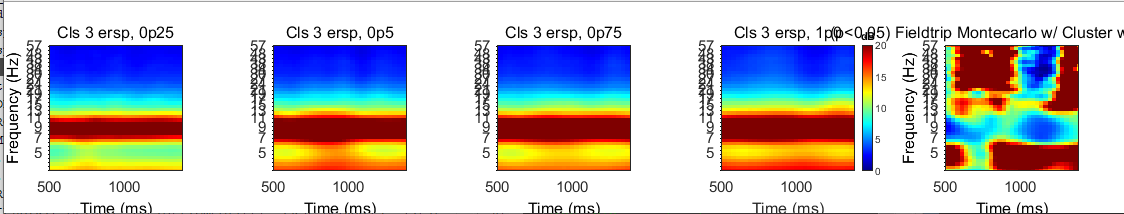
### TEST\_nobaseline

No time warp cropping



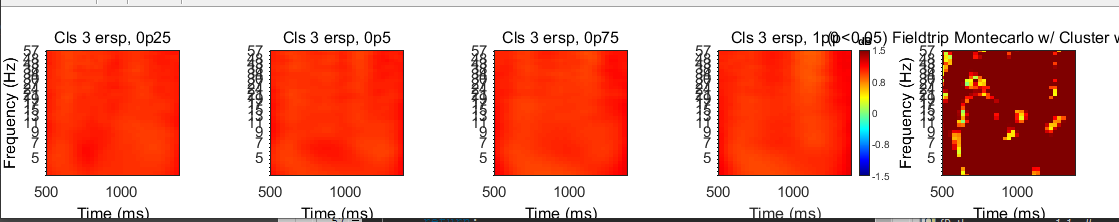
Time warp cropping





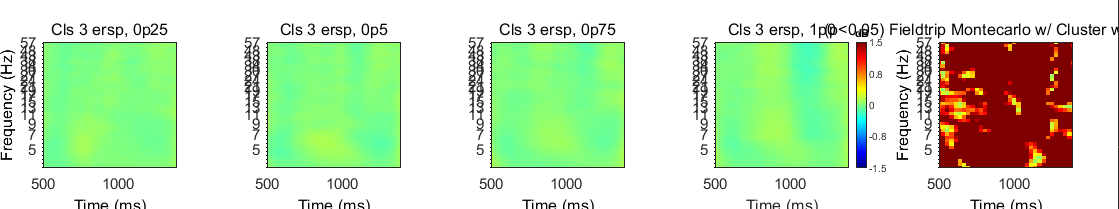
### TEST1

{NaN,’off’,[500ms,1415ms],’off’,’off’,’off’,’on’}

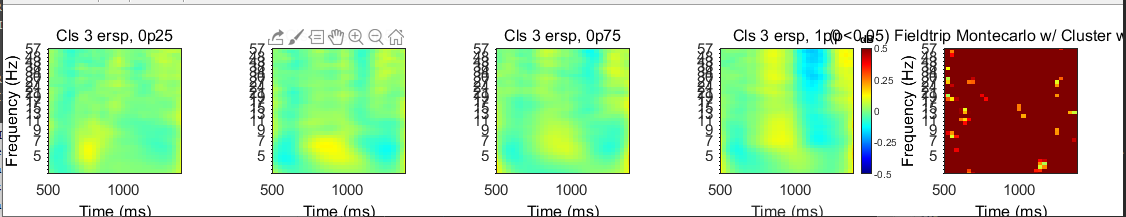


### TEST3

{NaN,’on’,[500ms,1415ms],’off’,’off’,’off’,’on’}

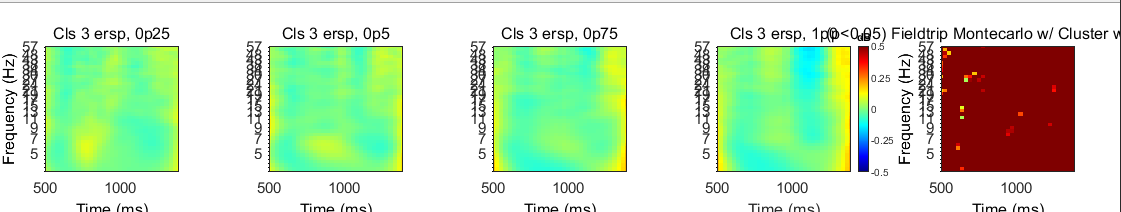


Changing color axis…



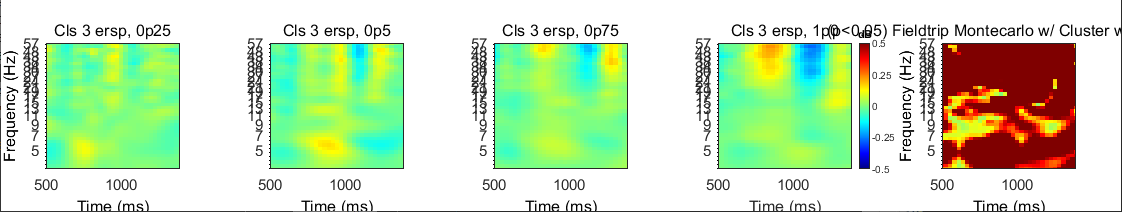
###TEST4

{NaN,’on’,[500ms,1415ms],’on’,’off’,’off’,’on’}



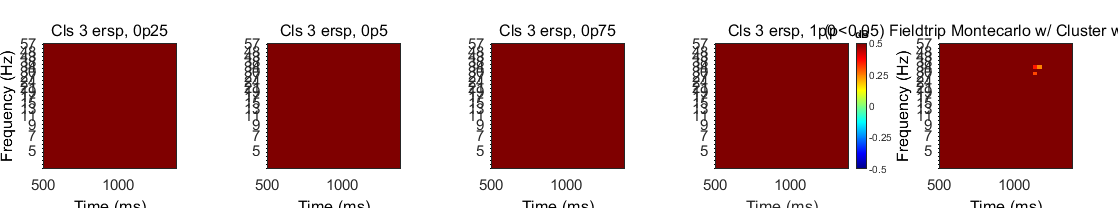
###TEST5

{NaN,’on’,[500ms,1415ms],’on’,’on’,’off’,’on’}



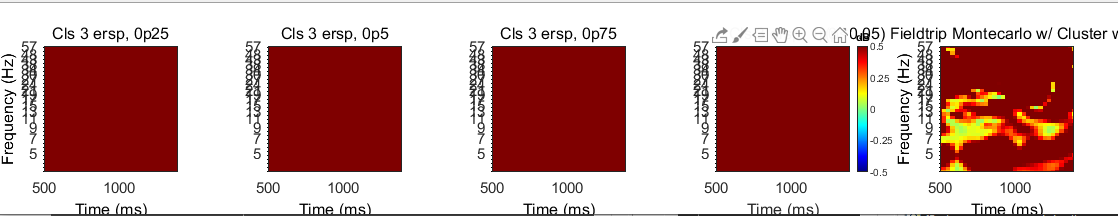
###TEST5

{NaN,’on’,[500ms,1415ms],’on’,’on’,’on’,’on’}



###TEST6

{NaN,’off’,[500ms,1415ms],’off’,’on’,’off’,’on’}



## Notes (Chang Liu)

* Average within person, otherwise, you would then be biasing towards an individual with more gait cycles for a particular condition
* [-1.2,1.2] senorimotor, and [-0.6,0.6] for the rest of the areas.
* Violinplot code: R:\Ferris-Lab\liu.chang1\Code\MiM\_HY\12\_behavioral\_analysis
* Try implementing countourf.m for plotting ERSP