Transient Beta Activity

Written: Jun 22, 2023, **Last updated:** Jul 3, 2023

Gerardo Parra

1. OVERVIEW

1.A Background

Oscillatory activity has traditionally been studied through average EEG power within a frequency band, across time and trials. For the beta band, recent studies suggest beta activity arises in the form of transient, “bursty” events, which only give the illusion of sustained oscillations when averaging across time and trials (**Figure 1**). Studying beta activity at the event level has revealed beta events modulate tactile perception: beta events preceding a stimulus inhibit perception, while those simultaneous to a stimulus enhance perception.

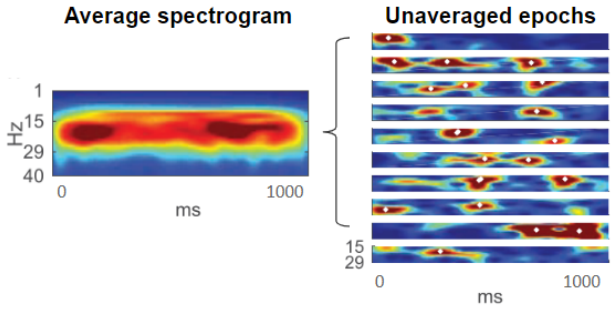

Figure 1. Averaging beta power obscures the transient nature of beta activity (Adapted from Shin et al., 2017)

Transient beta activity can be characterized by beta event rate, event duration, event frequency span, and event power. Among these features, event rate has been found to drive functionally meaningful differences in averaged beta power in primary somatosensory cortex. Computational modeling, MEG, and rodent studies suggest that beta events are generated by thalamocortical activity; namely, the “integration of nearly synchronous bursts of excitatory synaptic drive targeting proximal and distal dendrites of pyramidal neurons” (Law et al., 2022).

1.B In the Levin Lab

Given the relationship between beta events and sensory perception, we hypothesized transient beta activity may reflect endogenous/spontaneous brain activity which modulates the processing of sensory stimul1. This could support our view of sensory phenotypes in autism spectrum disorder (ASD) as potentially arising from a “leaky thalamus”.

We began by studying beta events in resting-state EEG data from SPA, and found that event rate and (intra-individual) event rate variability are altered in idiopathic ASD. Moreover, these features are associated with phenotypic measures of somatosensation and sensory seeking (SP2 and SAND). We found that event rate features are also altered in several neurogenetic disorders related to autism (Rett, PMS, SYNGAP).

To further understand the potential sources of these differences, we are examining transient beta activity in greater depth (e.g. topography, event waveforms, using subject-specific frequency bands). Moreover, we are studying beta events in SPA's sensory paradigms, to see how prestimulus beta events modulate evoked potentials.

1.C Literature

Make sure you have a solid understanding of the theoretical framework and experimental evidence underlying this analysis ***before*** *applying it to a dataset.* The following papers, sorted chronologically, trace the introduction of the concept of transient beta events up to the study of the laminar dynamics underlying them.

1. Sherman et al., *PNAS*, 2016, doi: 10.1073/pnas.1604135113
2. Shin et al., *eLife*, 2017, doi: 10.7554/eLife.29086
3. Bonaiuto et al., *NeuroImage*, 2021, doi: 10.1016/j.neuroimage.2021.118479
4. Law et al., *Cerebral Cortex*, 2022, doi: 10.1093/cercor/bhab221

2. ANALYZING TRANSIENT BETA EVENTS

The Jones Lab developed the SpectralEvents toolbox to detect and analyze transient oscillatory activity in M/EEG. We use a modified version of the toolbox to study beta events in our preprocessed data.

2.A Source code

- Modified SpectralEvents toolbox:
  neuro-levin\Public\EEG Analyses + Matlab scripts\transient beta\2023-07-03_Toolbox_GP
- Original toolbox (for reference): SpectralEvents Toolbox v0.1

2.B Overview of differences from the original toolbox

- Added conversion of BEAPP data into SpectralEvents format
- Re-structured data more conveniently
  - Analyze data by participant
  - Create structs containing detected events for each channel
  - Output in formats similar to BEAPP: .mat file for each participant containing data across channels
- Added script for aligning events events detected via time-frequency response (TFR) to time series
  - Necessary for any analyses of event waveforms, but original toolbox does not include this

2.C Running the pipeline

Running the pipeline is relatively simple, only a few parameters should be configured on an .mlx file. The pipeline consists of two main components: (1) conversion of BEAPP data into SpectralEvents format, (2) event detection. The steps below outline how to make a copy of the code and run it.

2.C.I Cloning the scripts

1. On the neuro-levin RCFS server share, navigate to
   Public\EEG Analyses + Matlab scripts\transient beta
2. Make a copy of the folder specified in the **path above**, rename it with the following format:
   [current date in YYYY-MM-DD]_Toolbox_Copy_[dataset you are working with, e.g. SPA]
3. Add this folder and subfolders to your MATLAB path

2.C.II Setting parameters & running

1. Open **load_paths.m**
   a. Configure the paths to your dataset. The important inputs to configure are:
      i. path.beapp_files
      ii. path.beapp_tag
      iii. path.out_tables

2. Open **batch_spectral_LL.mlx**
    a. Configure the parameters for the data conversion and for finding events
    b. The table below defines each parameter and gives a suggested setting or an example
    c. The first code section contains parameters for converting your BEAPP data to SpectralEvents format
    d. The second code section contains parameters for running the event finding pipeline on the converted data
    e. The third code section contains parameters for creating an output table, i.e. specifying which features to extract from participants' data
    f. Once you have set all parameters, click 'Run'

2.D Parameters

This section details suggested parameters for

1. Preprocessing event-related data
2. Preprocessing resting-state data
3. Running the event-finding pipeline

2.D.I Preprocessing: Event-related data

Parameters have not been standardized for event-related EEG data.

2.D.II Preprocessing: Resting-state data

A 'base' user inputs script with recommended setting for preprocessing via BEAPP can be found in the folder you cloned under

.\reference\example_BEAPP_user_inputs.m

In addition, the following parameter(s) must be adjusted according to your analysis/dataset:

Parameter	Use	Suggested setting/example
grp_proc_info.beapp_ica_additional_chans_lbls{1}	Define any channels beyond the 10-20 system to include in analysis. Suggested: Left S1: [FC3, CP3 , C1 , C5], Right S1: [FC4, CP4, C2 C6], Frontal: [AF3 F1 Afz AF4 F2], Posterior: [P1 P03 Poz P04 P2]	For HCGSN 128: [29 35 41 42 37 30 105 111 110 103 93 87 23 19 16 4 3 1 60 67 72 77 85]

2.D.III Event-finding pipeline

The table below details each parameter's use and suggested setting.

Parameter	Use	Suggested setting/example
C.rerun	1: rerun the data conversion script on the files in the folder you specify below, 0: do not rerun the data conversion (skips to running the event-finding toolbox)	1 if running for the first time or processing new data. 0 otherwise
C.Fs	sampling rate of your pre-processed data. The pipeline will pull this from the batch's grp_proc_info if left blank	whatever sampling rate your processed data has. double check whether your data was downsampled during preprocessing.
C.path	(this applies to toolboxes beginning with 2023-07-03. for earlier versions, skip to the next row) struct containing information about where to find data. the relevant fields are: - beapp_files: path to folder with BEAPP outputs - beapp_tag: BEAPP tag to load data from - out_tables: path to save output tables to	C.path = load_paths configure the load_paths scripts first as described to the left
C.path.in	full file path to the folder containing your dataset's BEAPP output folders	example: fullfile('X:','Groups','SPA','01_Data_Raw_Summary_Processed','EEG','Participant_Data','03_Processed_Data','02_Baseline')
C.tag	BEAPP run tag of the batch of segmented data you wish to analyze	example: '_2023-05_resting'
P.band.name	label for the frequency band you are analyzing	'beta'
P.band.range	1x2 double with the lower and upper bounds of the frequency range to analyze	[13 30]
P.fVec	vector of frequencies to analyze, within P.band.range	P.band.range(1):4:P.band.range(end)
P.method	event finding method to use (1, 2, or 3) 1. maximal overlap (used in Shin et al., 2017) 2. minimal overlap 3. minimal overlap, event guarantee	2
P.ncyc	number of cycles to use in wavelets for TFR	5

Parameter	Use	Suggested setting/example
P.tag	run tag for toolbox outputs. different from C.tag: C.tag will be used to create a BEAPP module-like output folder. Within this folder, P.tag can be used to differentiate SpectralEvents batches run with different parameters (e.g. event-finding method, wavelet cycles)	today's date in YYYY-MM-DD
P.vis	1: output visualizations generated by the toolbox (will significantly increase computation time), 0: no visualizations	0
F.channels	channels to extract features from (separately, not averaged)	C3 and C4 for HCGSN 128: [36 104]
F.features	features to extract from each channel	{'rate','power','duration','ts_duration','Fspan'};
F.id_pat	string pattern to match to in subject filenames to extract ID	e.g. if filenames have the format '####_paradigm_name', digitsPattern(4) would pull the four digits at the beginning of the filename

2.E Outputs

- The pipeline will create a folder *spectral_events_[C.tag]* in the directory *C.path.in* which will contain all outputs from the pipeline, including converted data and spectral events analysis
- The pipeline will create an output table with the desired features for each participant in *C.path.out_tables*

2.E.I Converted data

- Converted BEAPP files will be saved in the output folder under *inputs*
- One .mat file containing converted data will be created for each participant/.mat file in *segment_[C.tag]*
- Each .mat file will contain the following variables
    - C - struct containing the parameters the conversion was run with
    - X - 1xN_chan struct (N_chan = number of EEG channels in data) with the following fields:
        - data - converted BEAPP data in SpectralEvents format
        - class - array indicating experimental condition/category for each segment in the data
    - file_proc_info - the same as in BEAPP

2.E.II Event-finding results

- Post-processed data will be saved in the output folder under the folder *[P.band.name]_[P.band.range(1)]to[P.band.range(end)]_[epoch time range]_[P.tag]_method[P.method]*
- One .mat file will be created for each participant/converted file
- Each .mat file will contain the following variables
    - C - struct containing the parameters the conversion was run with

- P - struct containing the parameters the event-finding pipeline was run with
- SE - 1xN_chan struct (N_chan = number of EEG channels in data) with the following fields:
  - trials - trial-level features
  - events - event-level features
  - IEI - inter-event interval features
- TFR - struct containing time-frequency response, in factors of median. Contains the following fields
  - channel - channel name, e.g. 'C3'
  - tfr - time frequency response
- X - struct containing converted data, as described above

2.E.II Output tables

- Output tables will have one row for each participant, and columns according to the features extracted:
- For each channel in *F.channels*
  - For each feature in *F.features*
    - The mean and standard deviation of the feature will be saved

3. USERS, UPDATES, and TROUBLESHOOTING

3.1 Users and Uses

New users of this analysis/code should be added here, along with their purpose(s) and any notes. The point person should be notified, and will work to integrate any necessary updates to documentation and code base

User	Purpose (e.g. study)	Notes, links to source code branches
Gerardo Parra	- SPA - Beta event Manuscript 1	

Klara Szilagyi	- Beta event Manuscript 1
	  - Neurogenetics datasets
	  - ABCCT	

3.2 Updates and Troubleshooting

If there are any issues (e.g. missing helper scripts in the source code, mysterious errors, etc), log them here and assign the problem to this toolbox's point person

Also log any planned, in-progress, or completed updates to the source code

Issue	Status	Notes/Person who completed it
Conversion of pipeline into BEAPP module	Not started	Gerardo Parra
Standardization of preprocessing parameters for event-related data	Not started	Yael Braverman (?)
Pipeline revamp	Approved	Gerardo Parra - align TFR events to time series - output format: .mat file per participant, struct for each channel, structs containing batch parameters