

# PyBispectra

## An open-source toolbox for advanced electrophysiological signal processing based on the bispectrum

Thomas S. Binns<sup>1,2</sup>, Franziska Pellegrini<sup>3</sup>, Tin Jurhar<sup>3</sup>, Stefan Haufe<sup>2,3</sup>

1. Movement Disorders Unit, Charité - Universitätsmedizin Berlin, Germany

2. Einstein Center for Neurosciences Berlin, Charité - Universitätsmedizin Berlin, Germany

3. Electrical Engineering and Computer Science Department, Technische Universität Berlin, Germany

### BACKGROUND

- Phase-amplitude coupling, time delays, and non-sinusoidal waveform features provide insights into neuronal function and dysfunction.
- Common methods for these analyses possess notable limitations.
- The bispectrum – the Fourier transform of the third order moment – can quantify these features without many of the limitations.
- We present PyBispectra, an open-source Python-based toolbox for analysing electrophysiological signals using the bispectrum.

The bispectrum has the fundamental form

$$\mathbf{B}_{kmn}(f_1, f_2) = \langle \mathbf{k}(f_1) \mathbf{m}(f_2) \mathbf{n}^*(f_2 + f_1) \rangle$$

where  $\mathbf{B}$  is the bispectrum;  $kmn$  is a combination of signals with Fourier coefficients  $\mathbf{k}$ ,  $\mathbf{m}$ , and  $\mathbf{n}$ , respectively;  $f_1$  and  $f_2$  correspond to a lower and higher frequency, respectively; and  $\langle \rangle$  represents the average value over epochs.

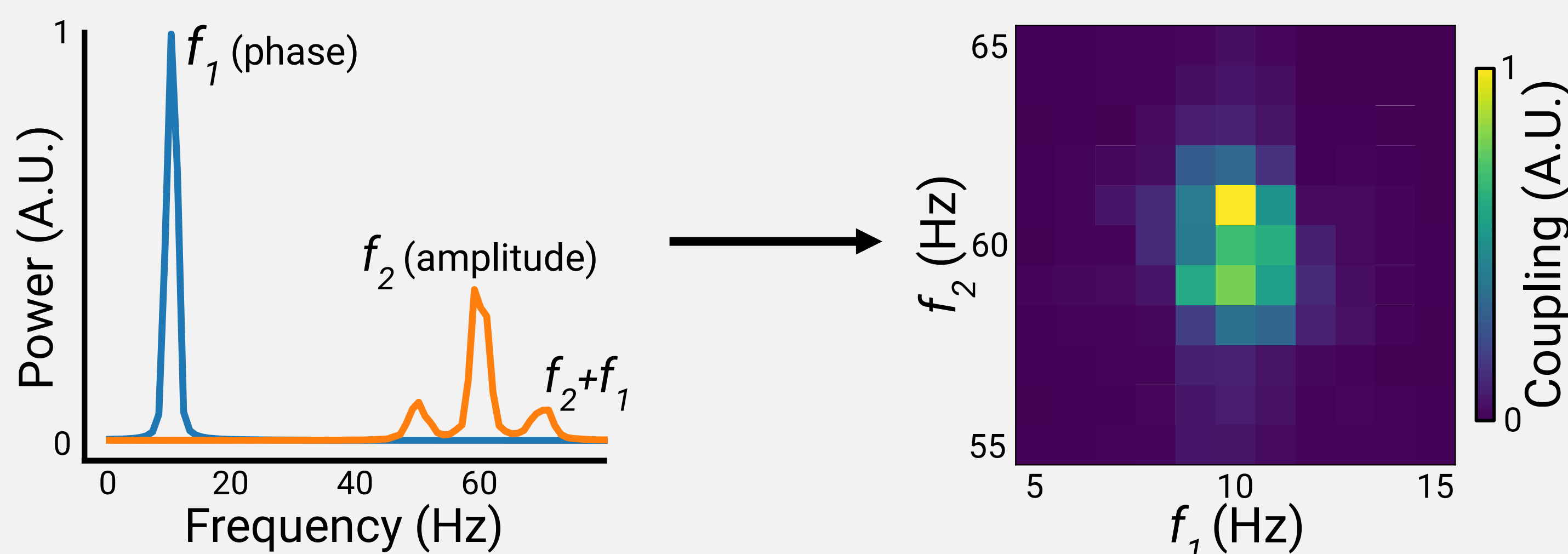
### USE CASE: PHASE-AMPLITUDE COUPLING [1, 2]

**Motivation:** mechanism for integrating information between neuronal systems across spatiotemporal scales, disrupted in neurological disorders.

**Common method:** modulation index – requires the use of precise, difficult to find filters; relies on the computationally-expensive Hilbert transform; compromised by volume conduction (spurious across-site coupling estimates).

**Bispectrum:** no filters required; relies on the computationally-cheap Fourier transform; immune to volume conduction artefacts with antisymmetrisation.

#### Example: 10-60 Hz phase-amplitude coupling



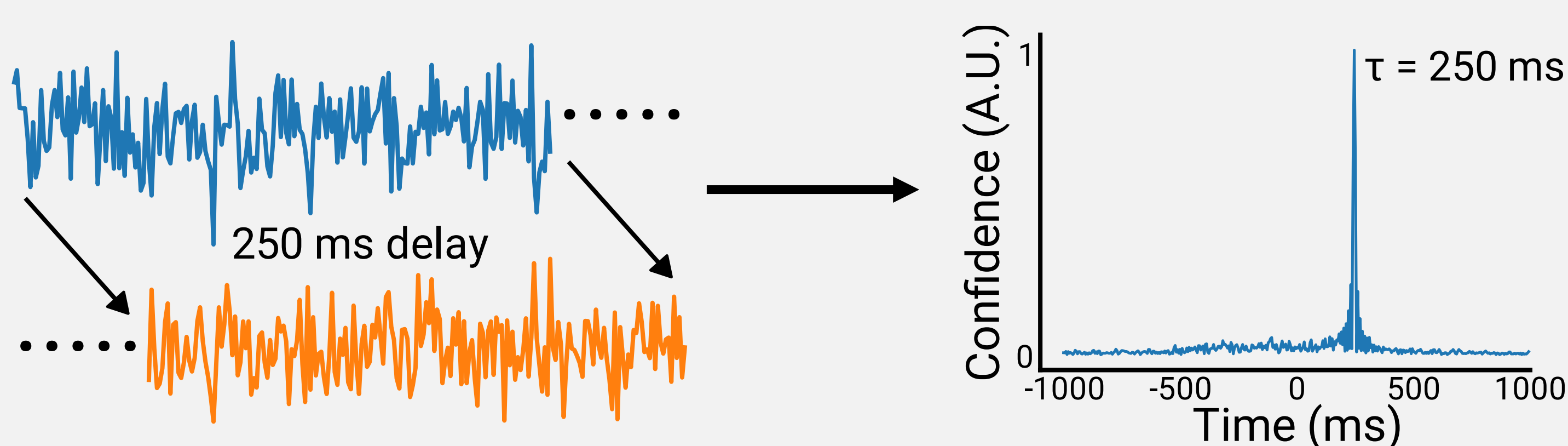
### USE CASE: TIME DELAY ESTIMATION [3, 4]

**Motivation:** reflects the physical connections between brain regions (e.g. projection distance, number of synapses, etc...).

**Common method:** cross-correlation – susceptible to Gaussian noise; compromised by volume conduction (spurious zero time-lag estimates).

**Bispectrum:** resilient to Gaussian noise; immune to volume conduction artefacts with antisymmetrisation.

#### Example: 250 ms time delay



References: [1] Zandvoort & Nolte (2021). Defining the filter parameters for phase-amplitude coupling from a bispectral point of view. *Journal of Neuroscience Methods*; [2] Pellegrini et al. (Pre-print). Distinguishing between- from within-site phase-amplitude coupling using antisymmetrized bispectral. *bioRxiv*; [3] Nikias & Pan (1988). Time delay estimation in unknown Gaussian spatially correlated noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing*; [4] Jurhar et al. (In Preparation). Estimating signal time-delays under mixed noise influence with novel cross- and bispectrum methods; [5] Bartz et al. (2019). Analyzing the waveshape of brain oscillations with bicoherence. *NeuroImage*.

### PyBispectra Pipeline

#### 0. Preprocessing (Optional)

```
from pybispectra import SpatioSpectralFilter
```

Enhance the signal-to-noise ratio for a frequency band of interest.

#### 1. Compute Frequency Information

```
from pybispectra import compute_fft, compute_tfr
```

Compute the (time-)frequency representation of data.

#### 2. Compute Results

```
from pybispectra import PAC, PPC, AAC, TDE, WaveShape
```

Support for cross-frequency coupling (phase-amplitude, phase-phase, amplitude-amplitude), time delay estimation, and waveshape analysis.

#### Example Code (Phase-Amplitude Coupling)

```
coeffs, freqs = compute_fft(data, sampling_freq)
pac = PAC(coeffs, freqs, sampling_freq)
pac.compute()
pac_results = pac.results
pac_results.plot()
```

Easy-to-use, with simple syntax

Detailed API documentation and tutorials

High performance

Check out the toolbox at:  
[pybispectra.readthedocs.io](https://pybispectra.readthedocs.io)



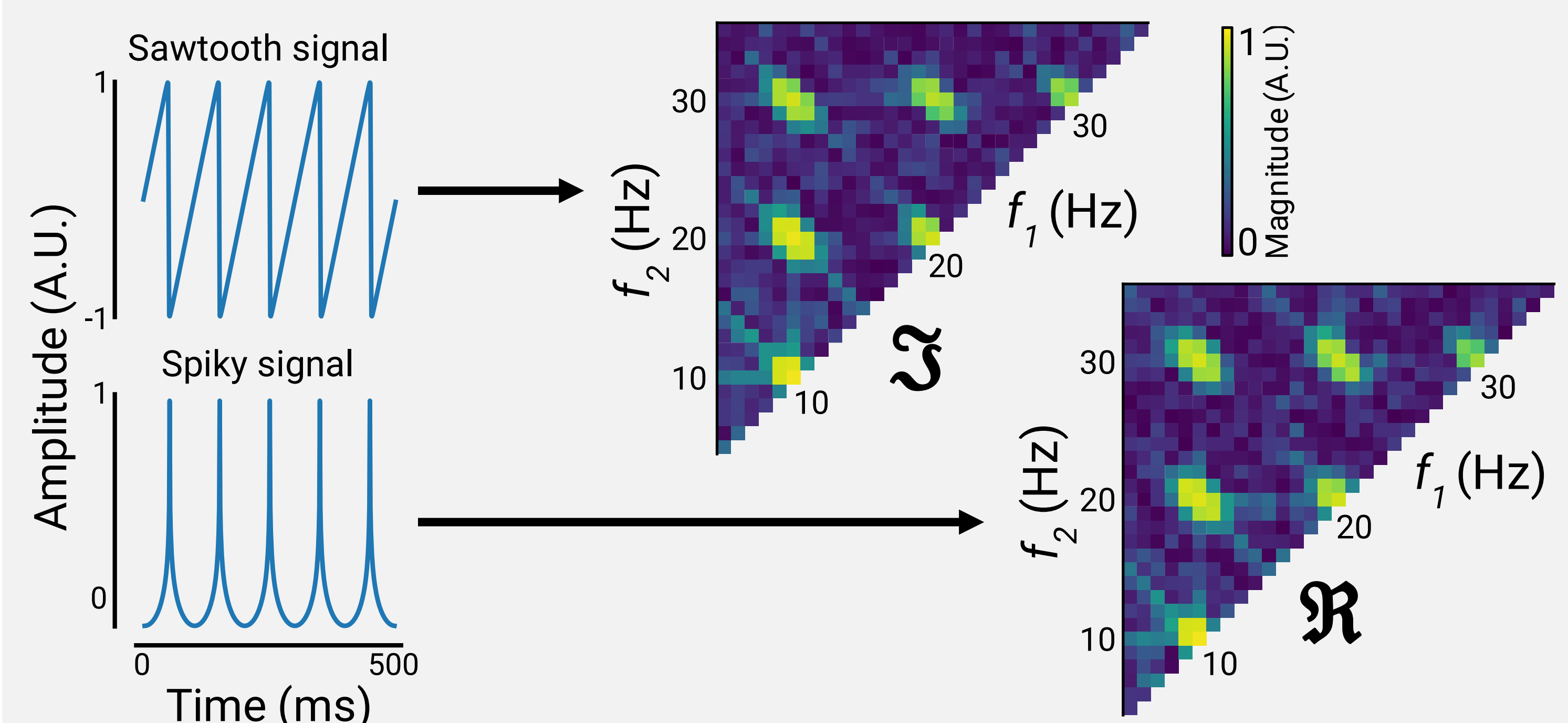
### USE CASE: WAVESHAPE ANALYSIS [5]

**Motivation:** non-sinusoidal features reflect synaptic input synchrony and are biomarkers of neurological disorders.

**Common method:** time-series analysis – necessary preprocessing steps (e.g. bandpass filtering) corrupt underlying signal shape; computationally expensive for high sampling rate data.

**Bispectrum:** extracts information about sawtooth (rise-decay asymmetry) and spike (peak-trough asymmetry) characteristics - provides a frequency-resolved result (no need to bandpass); computationally cheap, even for high sampling rate data.

#### Example: non-sinusoidal features at 10 Hz and harmonics



#### CONTACT

thomas-samuel.binns@charite.de  
haufe@tu-berlin.de

GitHub: /tsbinns

LinkedIn: /tsbinns

X: @tsbinns



Thomas S. Binns