

SIGNAL PROCESSING IN MNE: DAY 2

4 – EVOKED DATA THE EVOKED CLASS

08.02.2024

Thomas S. Binns

thomas-samuel.binns@charite.de



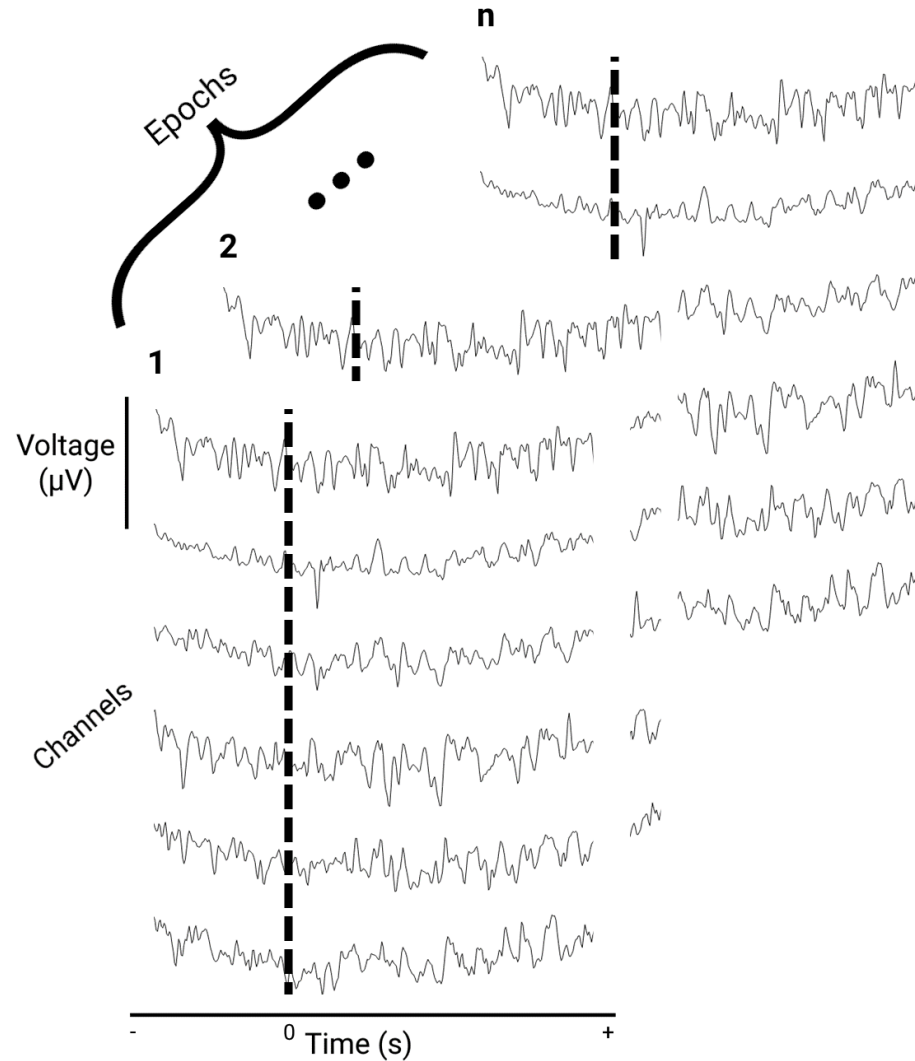
EINSTEIN
CENTER
Neurosciences



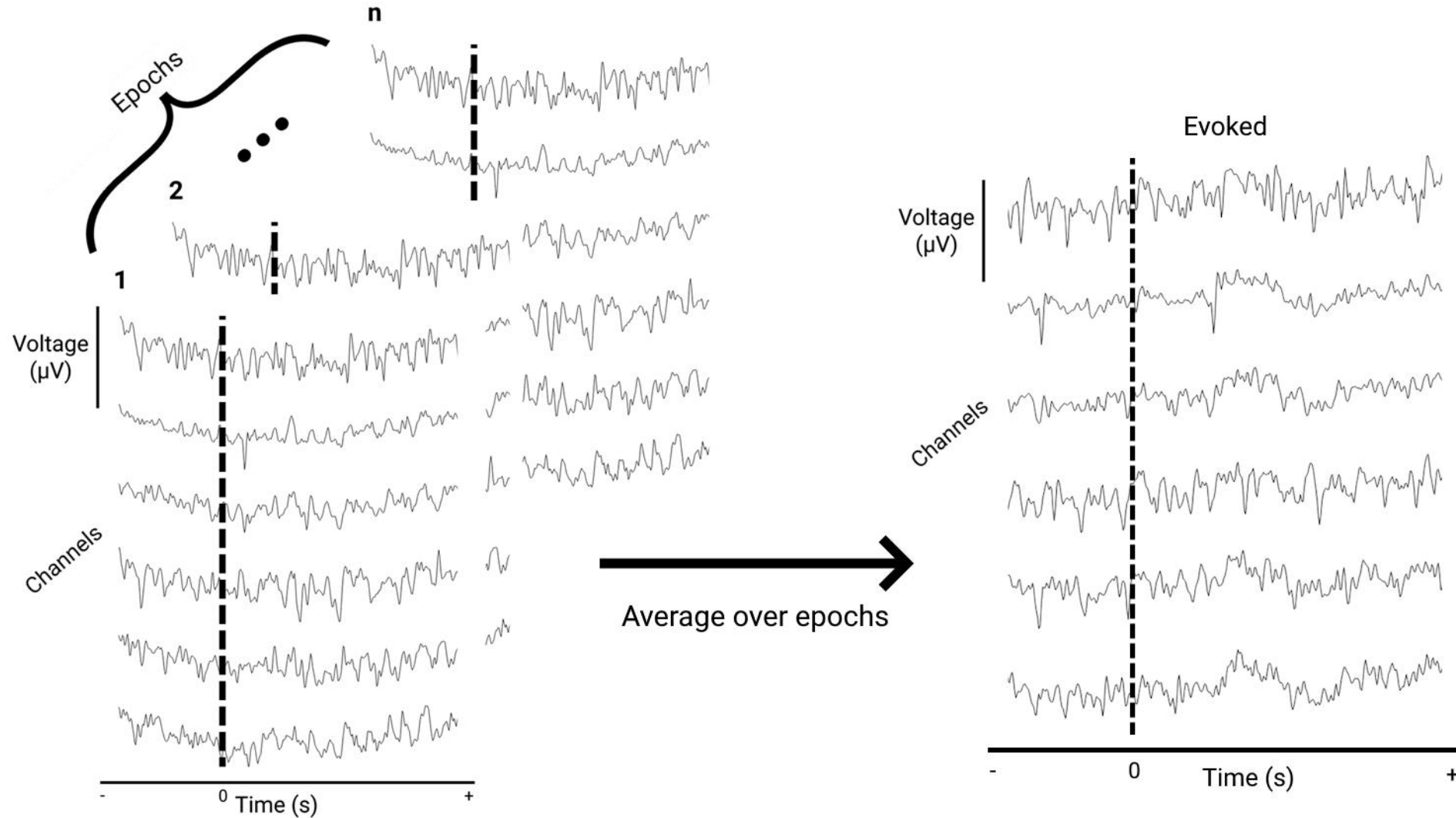
Brain and
Data Science
QAI LABS






Epoched data



Evoked data



Evoked data

MNE  [Install](#) [Documentation](#) [API Reference](#) [Get help](#) [Development](#)  

Section Navigation

- Most-used classes
 - [mne.io.Raw](#)
 - [mne.Epochs](#)
 - [mne.Evoked](#)**
 - [mne.Info](#)
- Reading raw data
- File I/O
- Creating data objects from arrays
- Exporting
- Datasets
- Visualization
- Preprocessing
- Events
- Sensor Space Data
- Covariance computation
- MRI Processing
- Forward Modeling
- Inverse Solutions

mne.Evoked

```
class mne.Evoked(fname, condition=None, proj=True, kind='average',
allow_maxshield=False, *, verbose=None)
```

[\[source\]](#)

Evoked data.

Parameters:

fname : *path-like*
Name of evoked/average FIF file to load. If None no data is loaded.

condition : `int`, or `str`
Dataset ID number (int) or comment/name (str). Optional if there is only one data set in file.

proj : *bool, optional*
Apply SSP projection vectors.

kind : `str`
Either `'average'` or `'standard_error'`. The type of data to read. Only used if 'condition' is a str.

allow_maxshield : *bool* | `str` (default `False`)
If True, allow loading of data that has been recorded with internal active compensation (MaxShield). Data recorded with MaxShield should generally not be loaded directly, but should first be processed using SSS/tSSS to remove the compensation signals that may also affect brain activity. Can also be `"yes"` to load without eliciting a warning.

Onto the notebook...

Conclusion

- Evoked data stored in `Evoked` & `EvokedArray` objects
- Can be created from `Epochs/EpochsArray` objects
`Epochs/EpochsArray` → `Evoked`
- Can be created from data arrays
`array (channels, times)` → `EvokedArray`
- Various methods for manipulating and visualising data stored in `Evoked` objects