In [1]:
```
# Dependencies
import mne
from antio import read_cnt
from antio.parser import read_triggers, read_info
import pandas as pd
import os
# import seaborn as sn

# Helper function
def set_use_evt(use_evt=True):
    if use_evt:
        try:
            # Restore the .EVT file
            os.rename('./evt_not_in_use.evt','./Subject_Example_2025-08-21_21-44-09
            print(r'Subject_Example_2025-08-21_21-44-09.evt restored')
        except:
            # The EVT file is already in place. No action needed
            print(r'The EVT file is already in place. No action taken')
    if not use_evt:
        try:
            # Hide the .EVT file by renaming it
            os.rename('./Subject_Example_2025-08-21_21-44-09.evt', './evt_not_in_us
            print(r'Subject_Example_2025-08-21_21-44-09.evt renamed as evt_not_in_u
        except:
            # The EVT file is already renamed/hidden. No action needed
            print(r'The EVT file is already renamed/hidden. No action taken')
```

In [2]:
```
# Load the data.....

# Using MNE with .EVT present
set_use_evt(use_evt=True)
mne_evt = mne.io.read_raw_ant('./Subject_Example_2025-08-21_21-44-09.cnt')
# Using antio with .EVT present
antio_evt = read_cnt('./Subject_Example_2025-08-21_21-44-09.cnt')

# Using MNE with .EVT absent
set_use_evt(use_evt=False)
mne_no_evt = mne.io.read_raw_ant('./Subject_Example_2025-08-21_21-44-09.cnt')
# Using antio with .EVT absent
antio_no_evt = read_cnt('./Subject_Example_2025-08-21_21-44-09.cnt')

set_use_evt(use_evt=True)
```

```
The EVT file is already in place. No action taken
Reading ANT file ./Subject_Example_2025-08-21_21-44-09.cnt
All 24 EEG channels are referenced to REF.
Subject_Example_2025-08-21_21-44-09.evt renamed as evt_not_in_use.evt
Reading ANT file ./Subject_Example_2025-08-21_21-44-09.cnt
All 24 EEG channels are referenced to REF.
Subject_Example_2025-08-21_21-44-09.evt restored
```

In [3]:
```
# Demonstrate that MNE and antio both show the same pattern.
# When the EVT file is present, it only sees the impedance markers from the beginni
# When the EVT file is absent, it misses the impedance markers but correctly reads
```

```
pd.concat({
    'MNE': pd.concat({
        'With EVT':pd.Series(read_triggers(antio_evt)[2]).value_counts(),
        'Without EVT':pd.Series(read_triggers(antio_no_evt)[2]).value_counts(),
    }, axis=1),

    'antio': pd.concat({
        'With EVT':mne_evt.annotations.to_data_frame()['description'].value_counts(
        'Without EVT':mne_no_evt.annotations.to_data_frame()['description'].value_c
    }, axis=1),
}, axis=1)
```

Out[3]:

	MNE		antio	
	With EVT	Without EVT	With EVT	Without EVT
impedance	2.0	NaN	2.0	NaN
0008	NaN	128.0	NaN	128.0
0009	NaN	32.0	NaN	32.0
0012	NaN	17.0	NaN	17.0
0011	NaN	16.0	NaN	16.0
0014	NaN	4.0	NaN	4.0
0013	NaN	4.0	NaN	4.0
0062	NaN	1.0	NaN	1.0
0063	NaN	1.0	NaN	1.0

In [4]:
```
# The timing is all fine. MNE is in ms or datetime, and antio is in samples, but th
df = pd.DataFrame.from_dict({
        'MNE':mne_evt.annotations.onset,
        'antio':read_triggers(antio_evt)[0],
    }, orient='columns')
df['ratio'] = df['antio']/df['MNE']
print('With EVT:')
df
```

```
With EVT:
```

Out[4]:

	MNE	antio	ratio
0	0.000	0	NaN
1	469.618	234809	500.0

In [5]:
```
# The timing is all fine. MNE is in ms or datetime, and antio is in samples, but th
df = pd.DataFrame.from_dict({
        'MNE':mne_no_evt.annotations.onset,
        'antio':read_triggers(antio_no_evt)[0],
    }, orient='columns')
```

```
df['ratio'] = df['antio']/df['MNE']
print('Without EVT:')
df
```

```
Without EVT:
```

Out[5]:

	MNE	antio	ratio
0	34.186	17093	500.0
1	34.596	17298	500.0
2	36.096	18048	500.0
3	38.130	19065	500.0
4	40.154	20077	500.0
...	...	...	...
198	421.286	210643	500.0
199	421.290	210645	500.0
200	441.304	220652	500.0
201	441.638	220819	500.0
202	443.000	221500	500.0

203 rows × 3 columns

In [6]:
```
# Only antio actually grabs the impedance values (there are 24 channels), and only
{'With EVT': [len(x) for x in read_triggers(antio_evt)[3]],
'Without EVT': [len(x) for x in read_triggers(antio_no_evt)[3]]}
```

Out[6]:
```
{'With EVT': [24, 24], 'Without EVT': []}
```

In [7]:
```
# All four methods do fine with channel names
pd.concat({
    'MNE':
    pd.DataFrame({'With EVT': mne_evt.ch_names,
                  'Without EVT': mne_no_evt.ch_names}),
    'antio':
    pd.DataFrame({'With EVT': read_info(antio_evt)[0],
                  'Without EVT': read_info(antio_no_evt)[0]}),
},axis=1)
```

Out[7]:

	MNE		antio	
	With EVT	Without EVT	With EVT	Without EVT
0	Fp1	Fp1	Fp1	Fp1
1	Fp2	Fp2	Fp2	Fp2
2	F9	F9	F9	F9
3	F7	F7	F7	F7
4	F3	F3	F3	F3
5	Fz	Fz	Fz	Fz
6	F4	F4	F4	F4
7	F8	F8	F8	F8
8	F10	F10	F10	F10
9	T9	T9	T9	T9
10	T7	T7	T7	T7
11	C3	C3	C3	C3
12	C4	C4	C4	C4
13	T8	T8	T8	T8
14	T10	T10	T10	T10
15	P9	P9	P9	P9
16	P7	P7	P7	P7
17	P3	P3	P3	P3
18	Pz	Pz	Pz	Pz
19	P4	P4	P4	P4
20	P8	P8	P8	P8
21	P10	P10	P10	P10
22	O1	O1	O1	O1
23	O2	O2	O2	O2