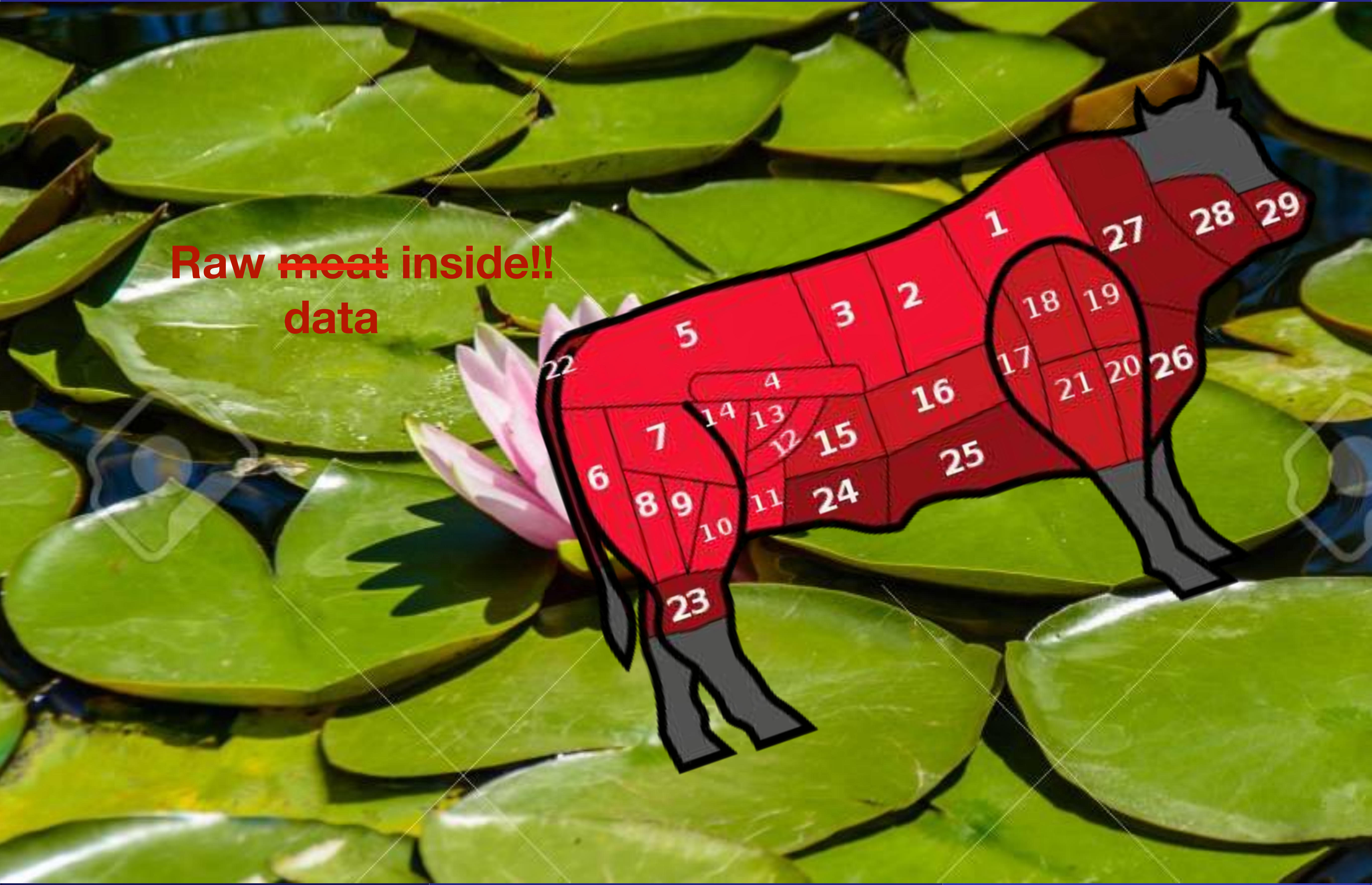
NenuFAR raw data with NenuRaw



Raw ~~meat~~ inside!!
data

NenuFAR waveform data with NenuRaw

Jump on your favourite Nancep machine

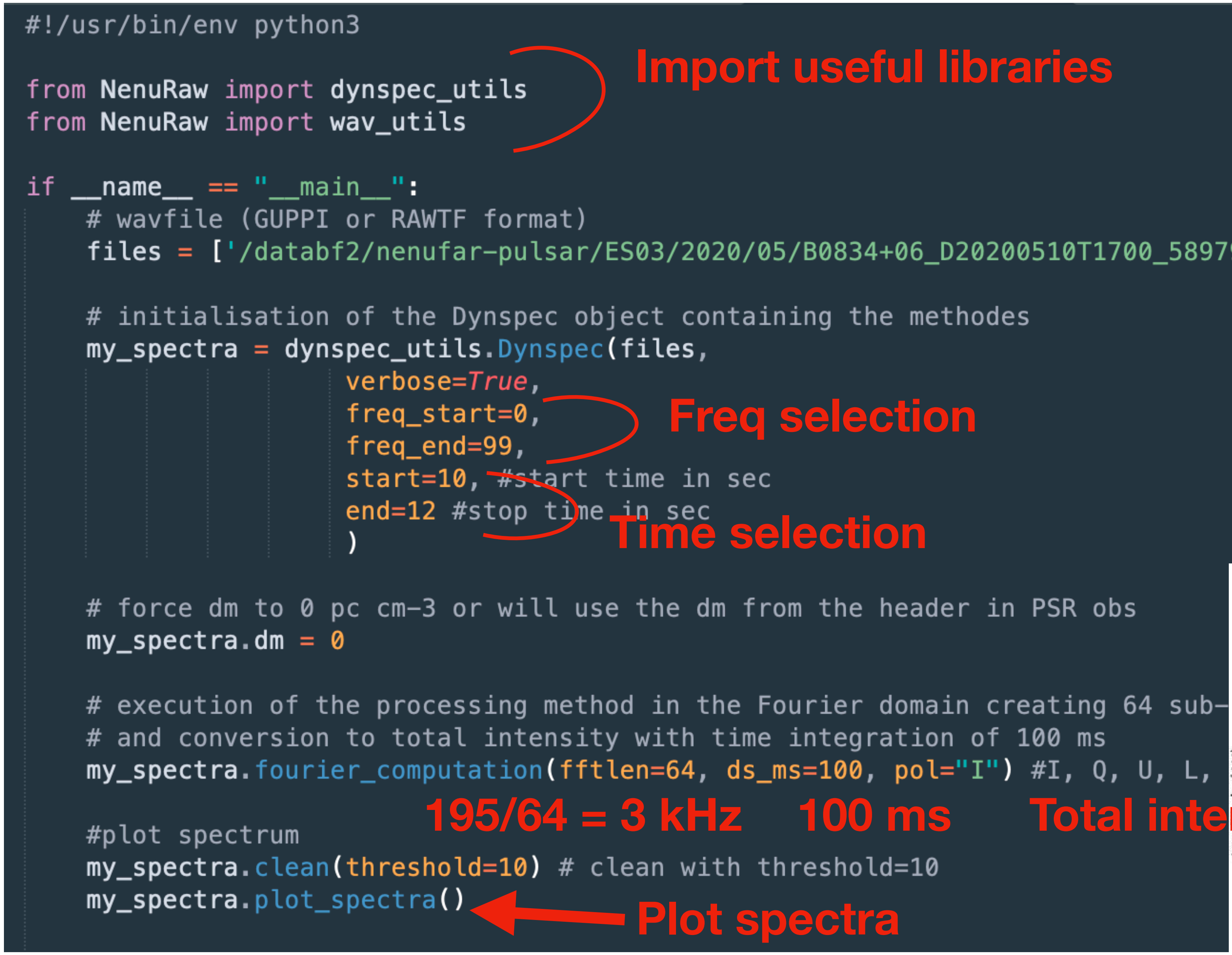
Source the python3.8 environment of NenuRaw

```
> source /home/lbondonneau/PSRpy3env/bin/activate
```

NenuFAR waveform data with NenuRaw (spectrum)

Python 3 script to plot the spectrum

```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_spectrum.py
```



```
#!/usr/bin/env python3

from NenuRaw import dynspec_utils
from NenuRaw import wav_utils

if __name__ == "__main__":
    # wavfile (GUPPI or RAWTF format)
    files = ['/databf2/nenufar-pulsar/ES03/2020/05/B0834+06_D20200510T1700_58979_250507_0069_BEAM1.0000.raw']

    # initialisation of the Dynspec object containing the methodes
    my_spectra = dynspec_utils.Dynspec(files,
                                       verbose=True,
                                       freq_start=0,
                                       freq_end=99,
                                       start=10, #start time in sec
                                       end=12 #stop time in sec
                                       )

    # force dm to 0 pc cm-3 or will use the dm from the header in PSR obs
    my_spectra.dm = 0

    # execution of the processing method in the Fourier domain creating 64 sub-
    # and conversion to total intensity with time integration of 100 ms
    my_spectra.fourier_computation(fftlen=64, ds_ms=100, pol="I") #I, Q, U, L,

    #plot spectrum
    my_spectra.clean(threshold=10) # clean with threshold=10
    my_spectra.plot_spectra()
```

Import useful libraries

Freq selection

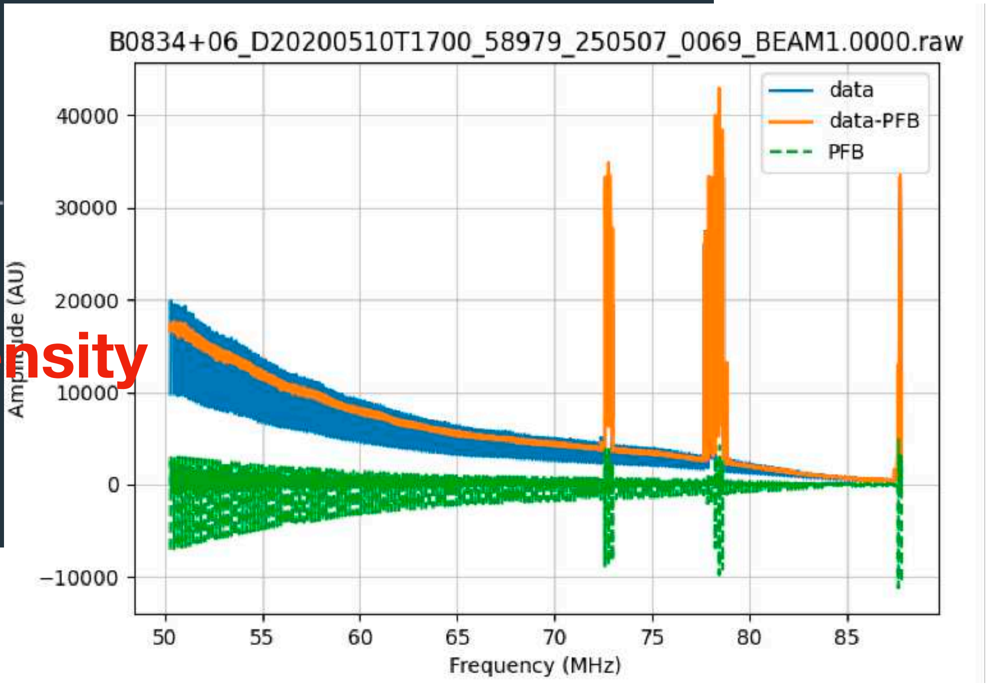
Time selection

195/64 = 3 kHz

100 ms

Total intensity

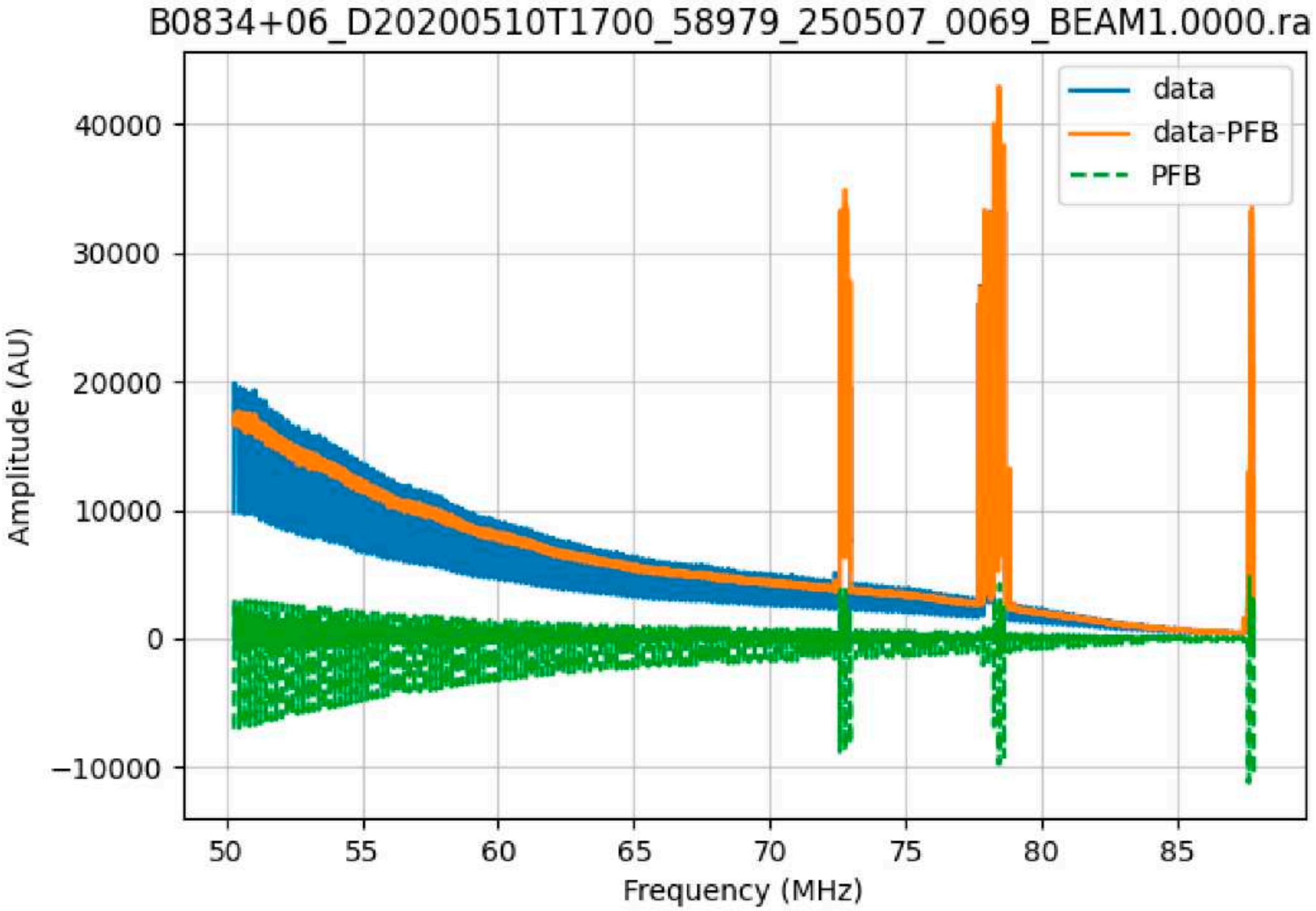
Plot spectra

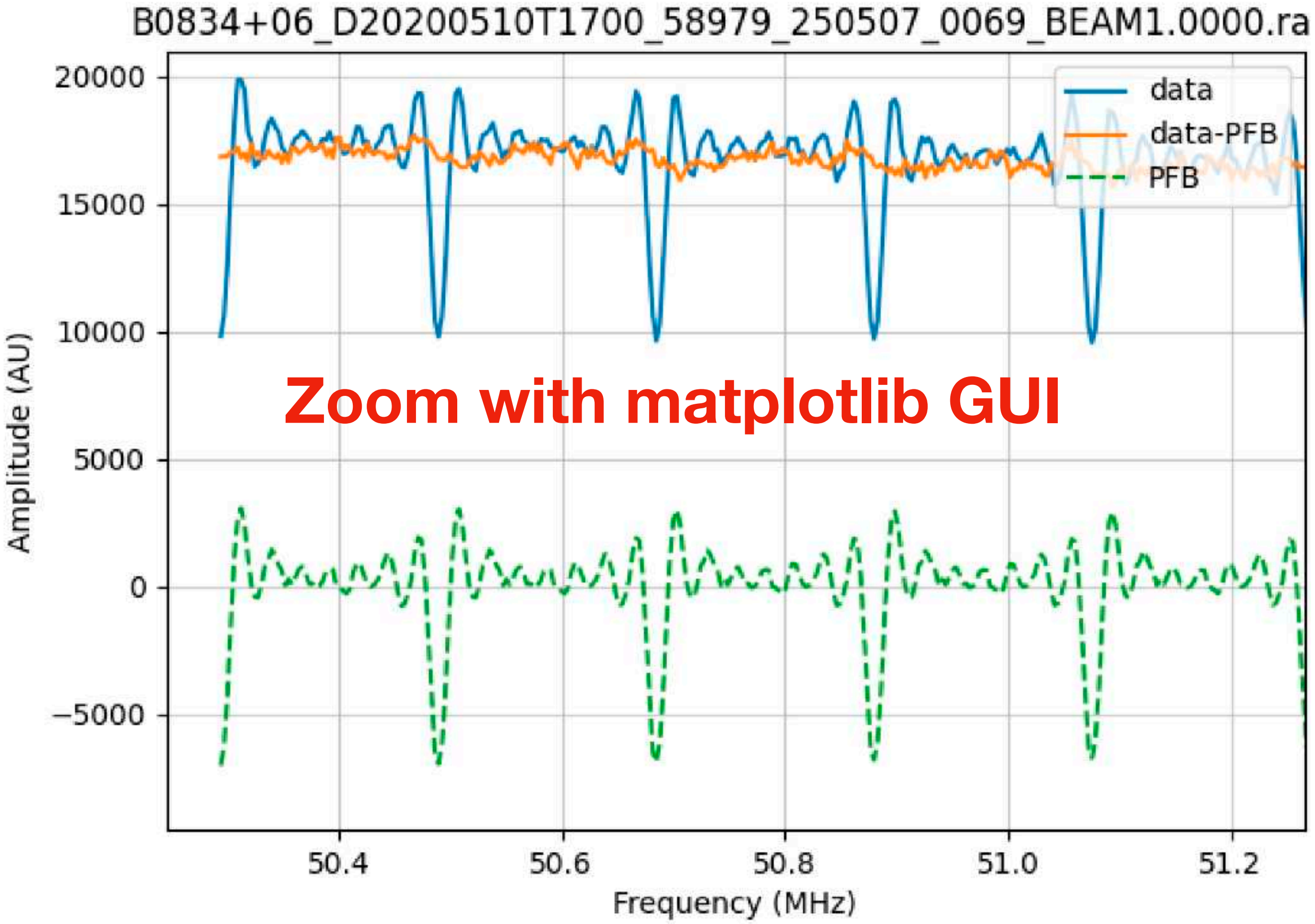


NenuFAR waveform data with NenuRaw (spectrum)

Python 3 script to plot the spectrum

```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_spectrum.py
```



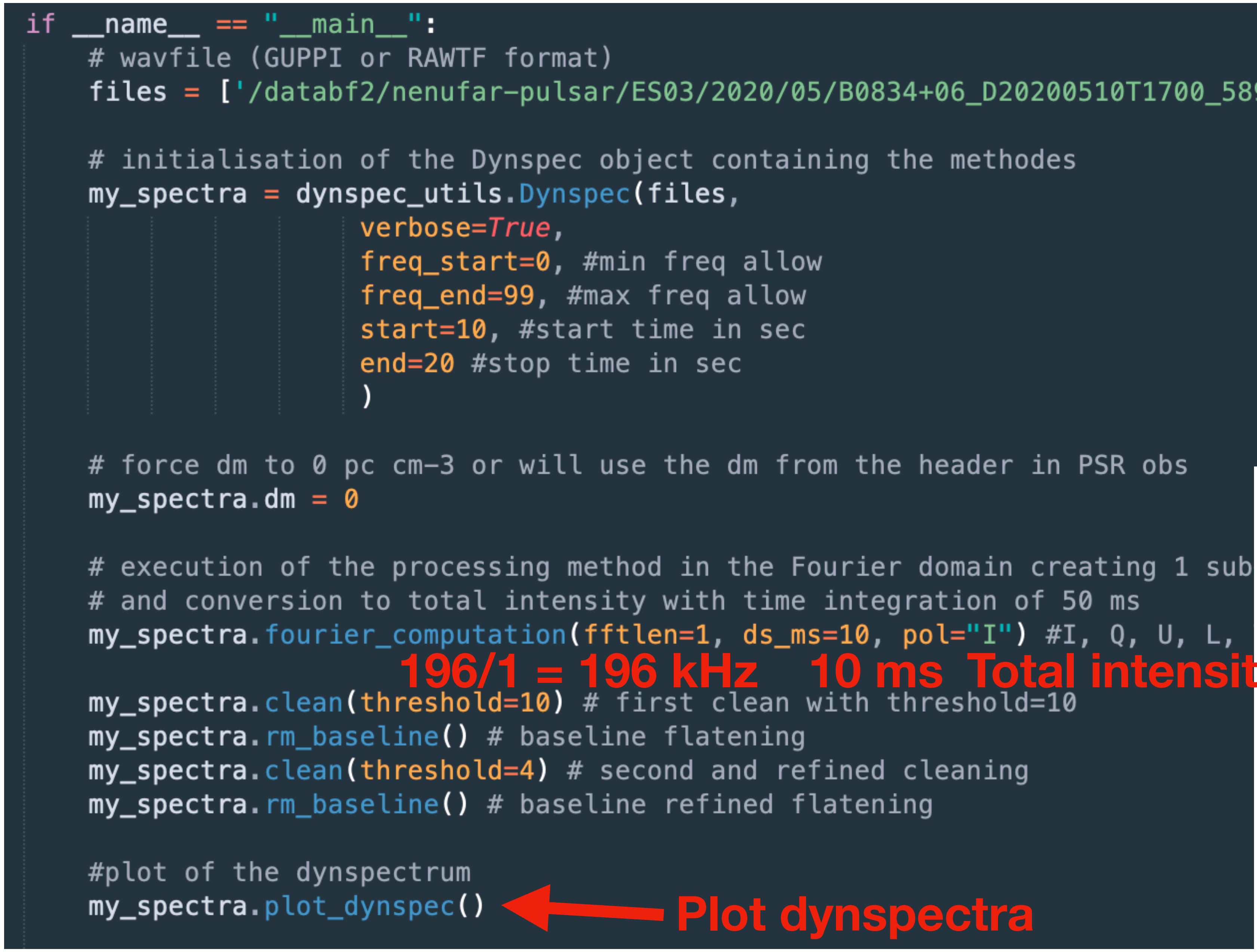


Zoom with matplotlib GUI

NenuFAR waveform data with NenuRaw (dynamic spectrum)

Python 3 script to plot the dynamic spectrum

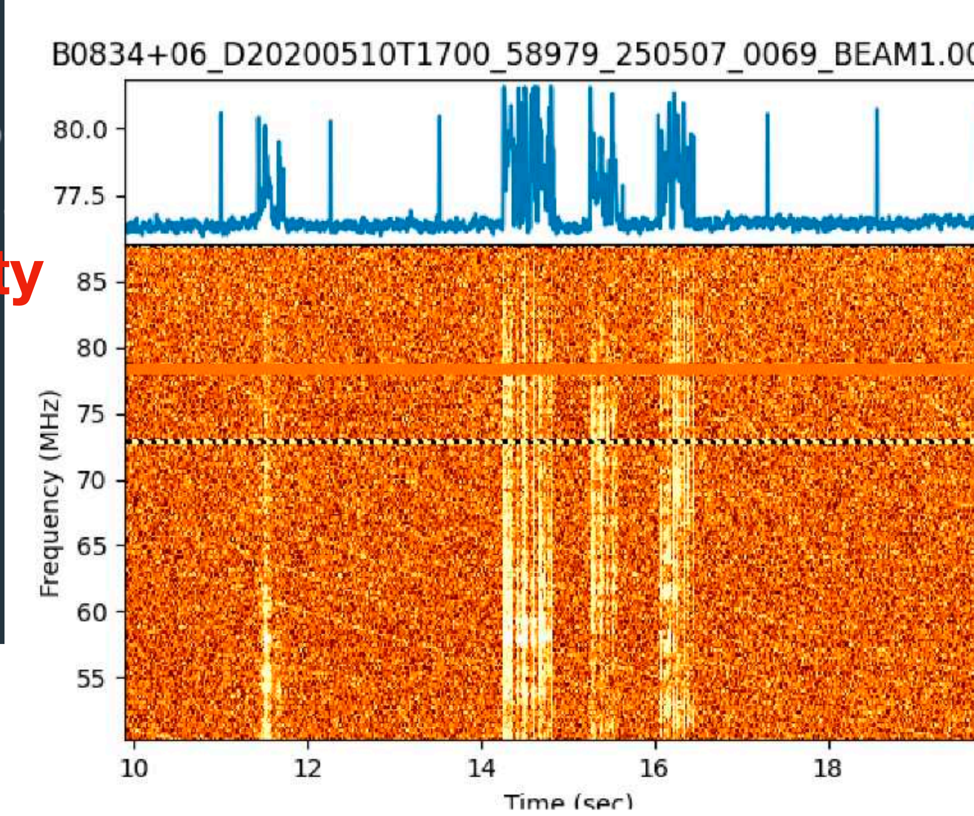
```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_storm_dynspectrum.py
```



```
if __name__ == "__main__":
    # wavfile (GUPPI or RAWTF format)
    files = ['/databf2/nenufar-pulsar/ES03/2020/05/B0834+06_D20200510T1700_58979_250507_0069_BEAM1.0000.raw']

    # initialisation of the Dynspec object containing the methodes
    my_spectra = dynspec_utils.Dynspec(files,
                     verbose=True,
                     freq_start=0, #min freq allow
                     freq_end=99, #max freq allow
                     start=10, #start time in sec
                     end=20 #stop time in sec
                     )

    # force dm to 0 pc cm-3 or will use the dm from the header in PSR obs
    my_spectra.dm = 0

    # execution of the processing method in the Fourier domain creating 1 sub
    # and conversion to total intensity with time integration of 50 ms
    my_spectra.fourier_computation(fftlen=1, ds_ms=10, pol="I") #I, Q, U, L,

    my_spectra.clean(threshold=10) # first clean with threshold=10
    my_spectra.rm_baseline() # baseline flatening
    my_spectra.clean(threshold=4) # second and refined cleaning
    my_spectra.rm_baseline() # baseline refined flatening

    #plot of the dynspectrum
    my_spectra.plot_dynspec()
```

196/1 = 196 kHz   10 ms   Total intensity

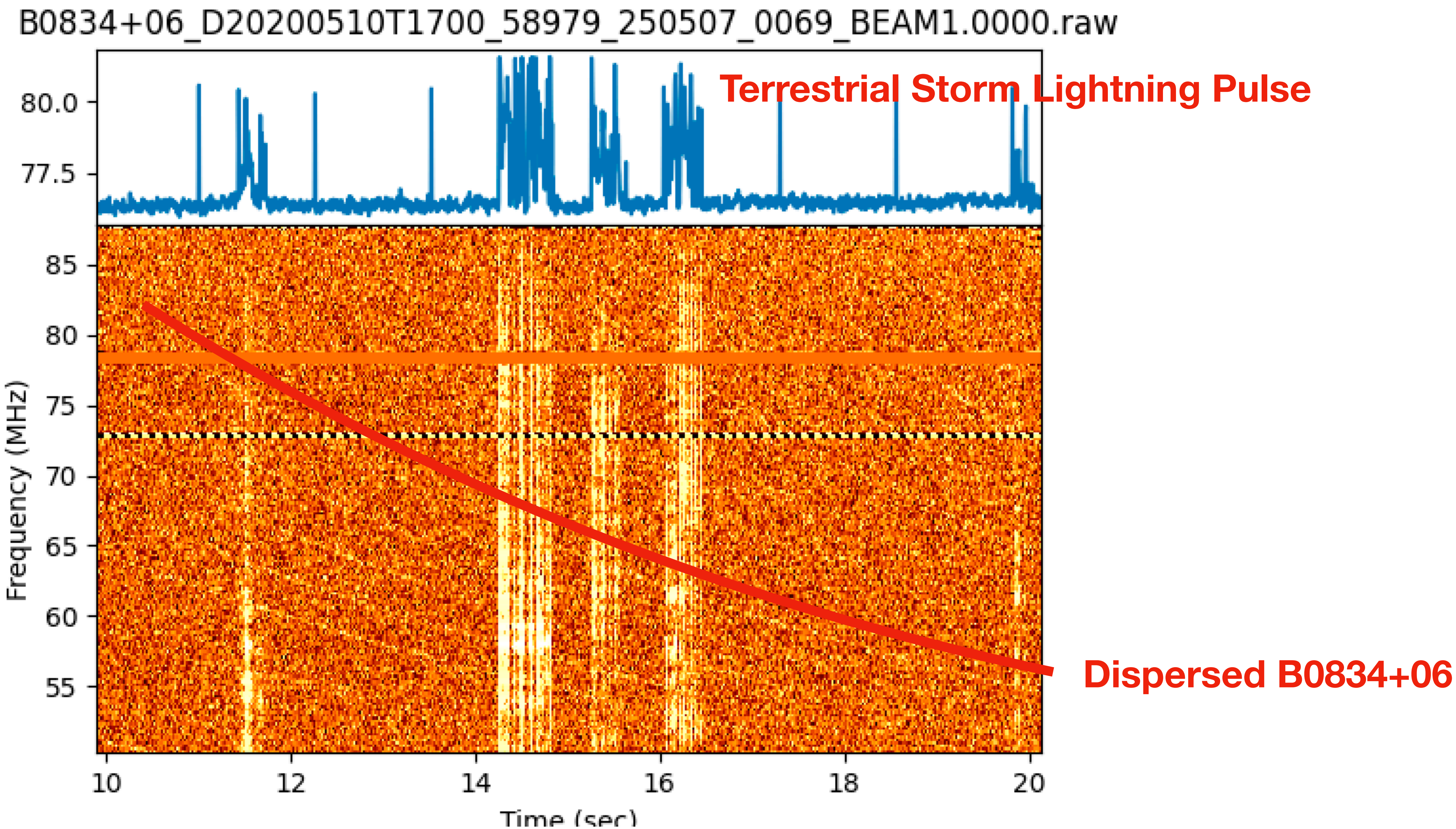
Plot dynspectra



NenuFAR waveform data with NenuRaw (dynamic spectrum)

Python 3 script to plot the dynamic spectrum

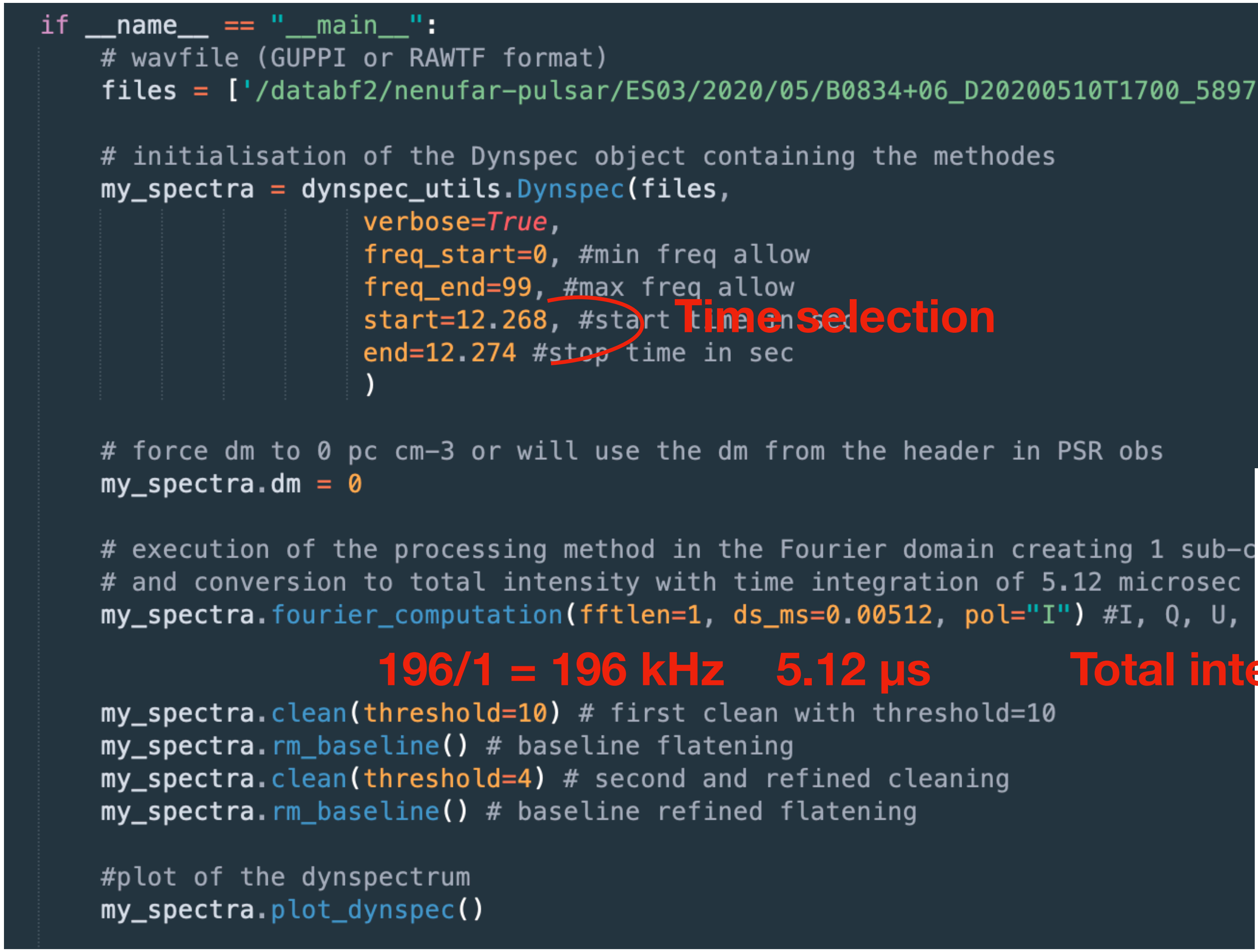
```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_storm_dynspectrum.py
```



NenuFAR waveform data with NenuRaw (dynamic spectrum)

Python 3 script to plot the dynamic spectrum

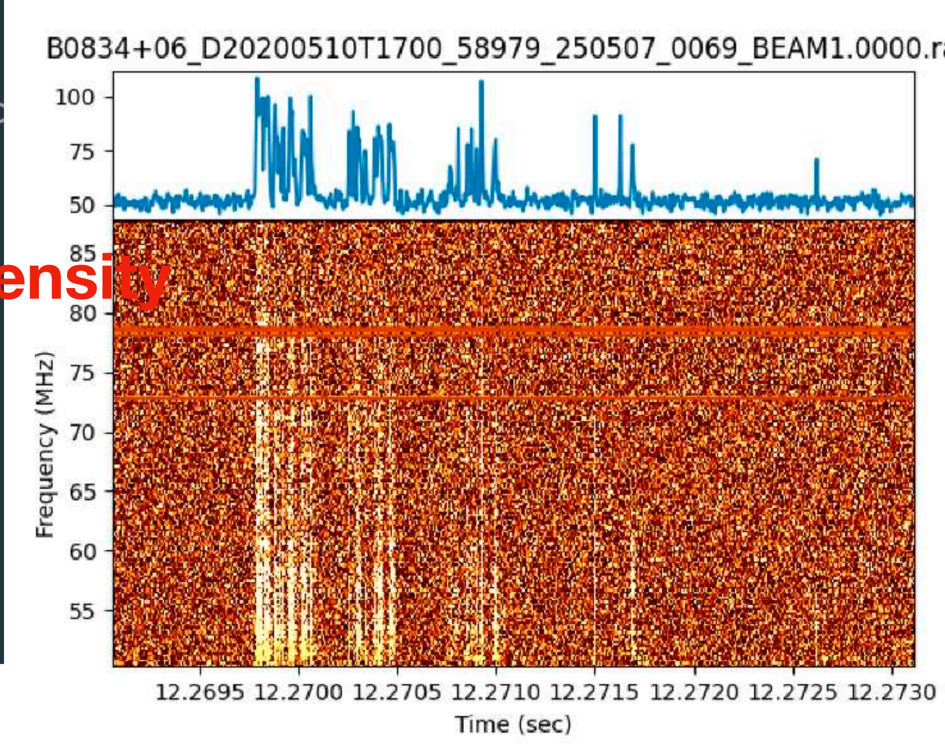
```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_storm_pulse.py
```



```
if __name__ == "__main__":
    # wavfile (GUPPI or RAWTF format)
    files = ['/databf2/nenufar-pulsar/ES03/2020/05/B0834+06_D20200510T1700_58979_250507_0069_BEAM1.0000.raw']

    # initialisation of the Dynspec object containing the methodes
    my_spectra = dynspec_utils.Dynspec(files,
                                       verbose=True,
                                       freq_start=0, #min freq allow
                                       freq_end=99, #max freq allow
                                       start=12.268, #start time in sec
                                       end=12.274 #stop time in sec
                                       )

    # force dm to 0 pc cm-3 or will use the dm from the header in PSR obs
    my_spectra.dm = 0

    # execution of the processing method in the Fourier domain creating 1 sub-c
    # and conversion to total intensity with time integration of 5.12 microsec
    my_spectra.fourier_computation(fftlen=1, ds_ms=0.00512, pol="I") #I, Q, U,

    my_spectra.clean(threshold=10) # first clean with threshold=10
    my_spectra.rm_baseline() # baseline flatening
    my_spectra.clean(threshold=4) # second and refined cleaning
    my_spectra.rm_baseline() # baseline refined flatening

    #plot of the dynspectrum
    my_spectra.plot_dynspec()
```

Time selection

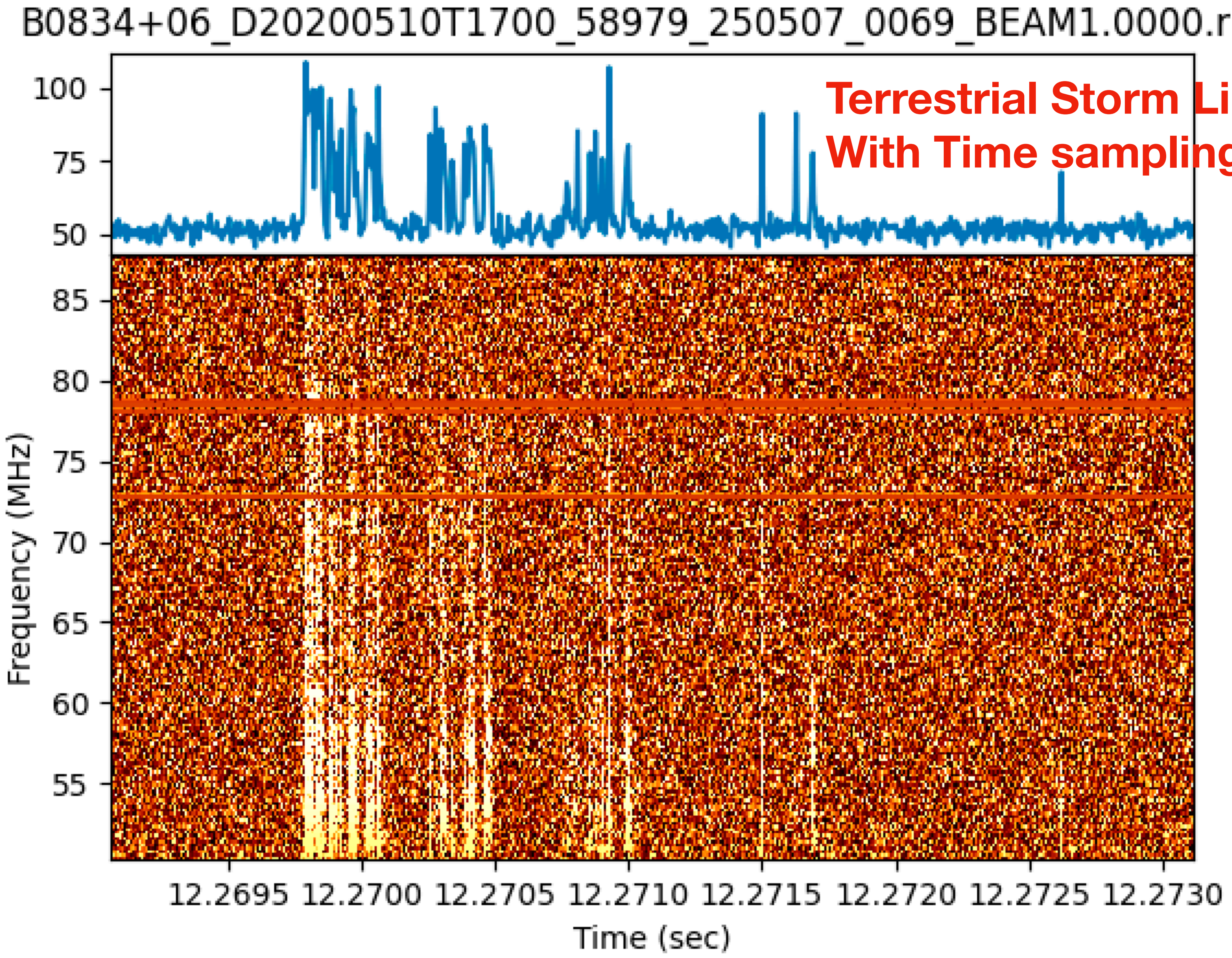
196/1 = 196 kHz   5.12 µs   Total intensity



NenuFAR waveform data with NenuRaw (dynamic spectrum)

Python 3 script to plot the dynamic spectrum

```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_storm_pulse.py
```

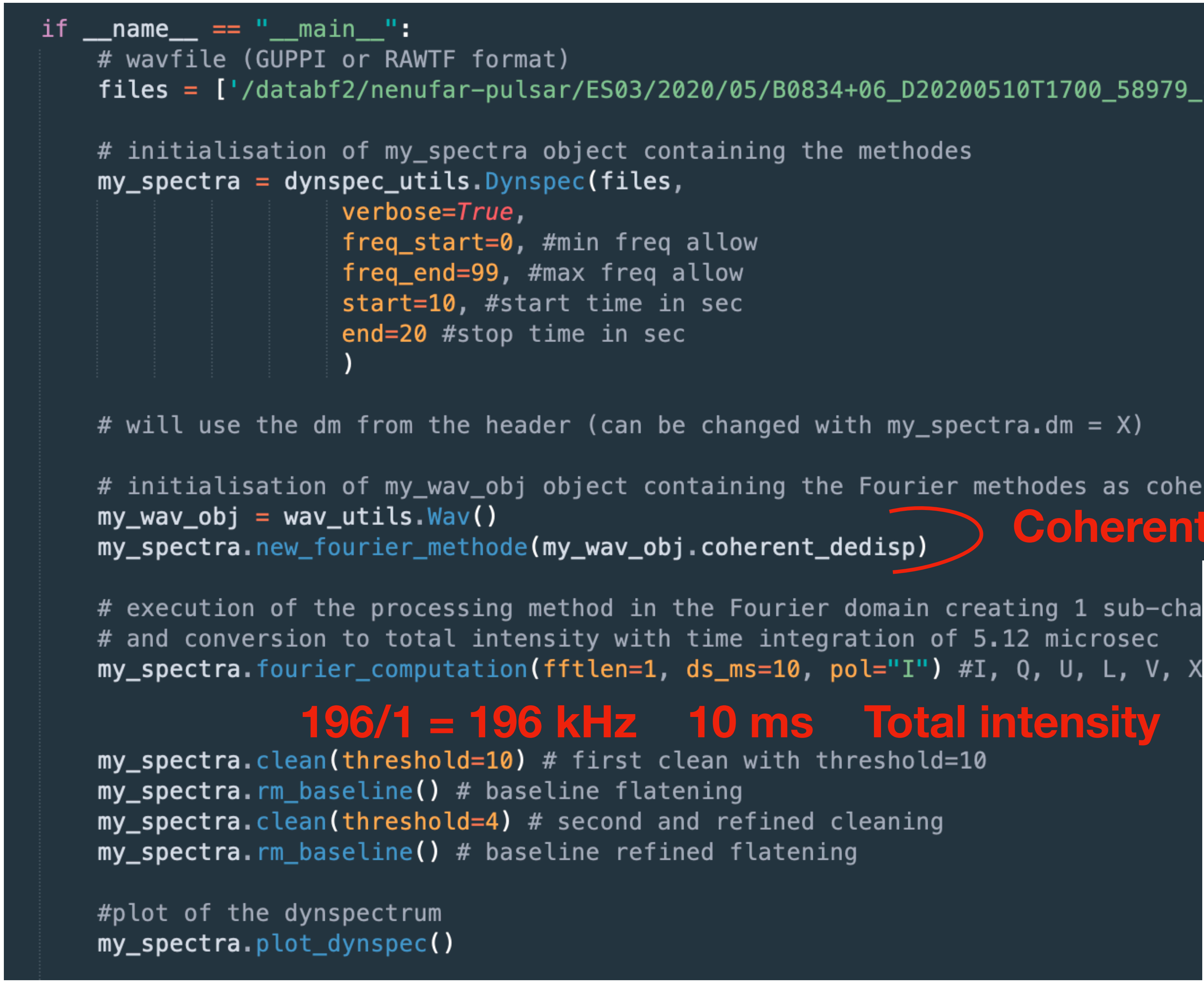


Terrestrial Storm Lightning Pulse
With Time sampling of 5.12 μs

NenuFAR waveform data with NenuRaw (dedisp dynspectrum)

Python 3 script to plot the dedispersed dynspec

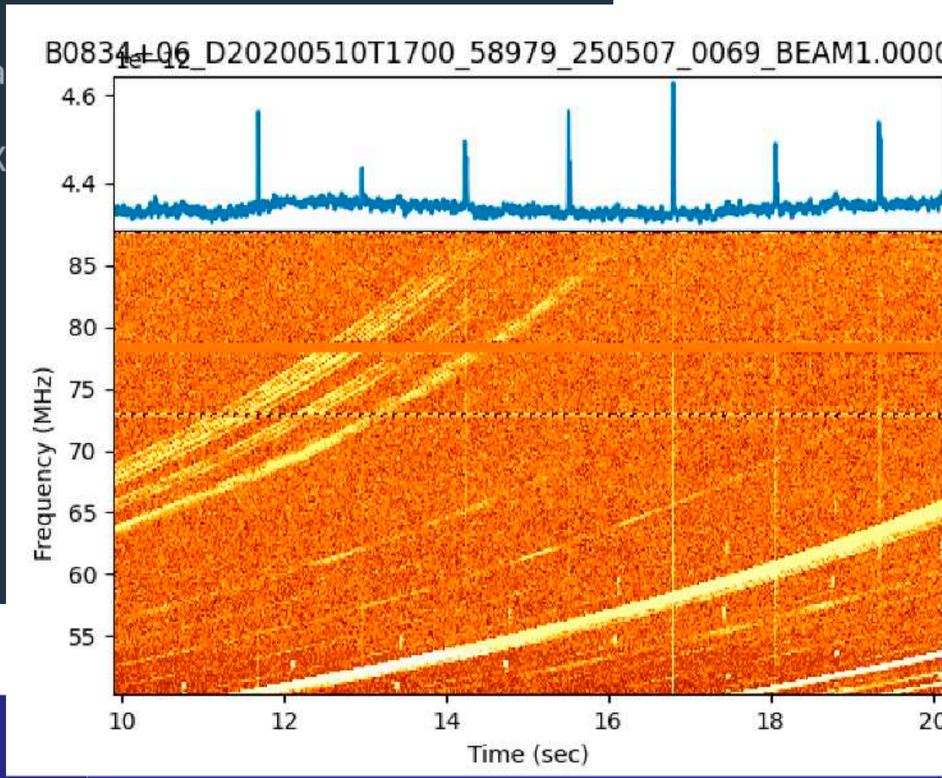
```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_B0834+06_dynspec.py
```



```
if __name__ == "__main__":
    # wavfile (GUPPI or RAWTF format)
    files = ['/databf2/nenufar-pulsar/ES03/2020/05/B0834+06_D20200510T1700_58979_250507_0069_BEAM1.0000.raw']

    # initialisation of my_spectra object containing the methodes
    my_spectra = dynspec_utils.Dynspec(files,
                                verbose=True,
                                freq_start=0, #min freq allow
                                freq_end=99, #max freq allow
                                start=10, #start time in sec
                                end=20 #stop time in sec
                                )

    # will use the dm from the header (can be changed with my_spectra.dm = X)

    # initialisation of my_wav_obj object containing the Fourier methodes as coherent dedispersion
    my_wav_obj = wav_utils.Wav()
    my_spectra.new_fourier_methode(my_wav_obj.coherent_dedisp)

    # execution of the processing method in the Fourier domain creating 1 sub-cha
    # and conversion to total intensity with time integration of 5.12 microsec
    my_spectra.fourier_computation(fftlen=1, ds_ms=10, pol="I") #I, Q, U, L, V, X

    my_spectra.clean(threshold=10) # first clean with threshold=10
    my_spectra.rm_baseline() # baseline flatening
    my_spectra.clean(threshold=4) # second and refined cleaning
    my_spectra.rm_baseline() # baseline refined flatening

    #plot of the dynspectrum
    my_spectra.plot_dynspec()
```

Coherent de-dispersion

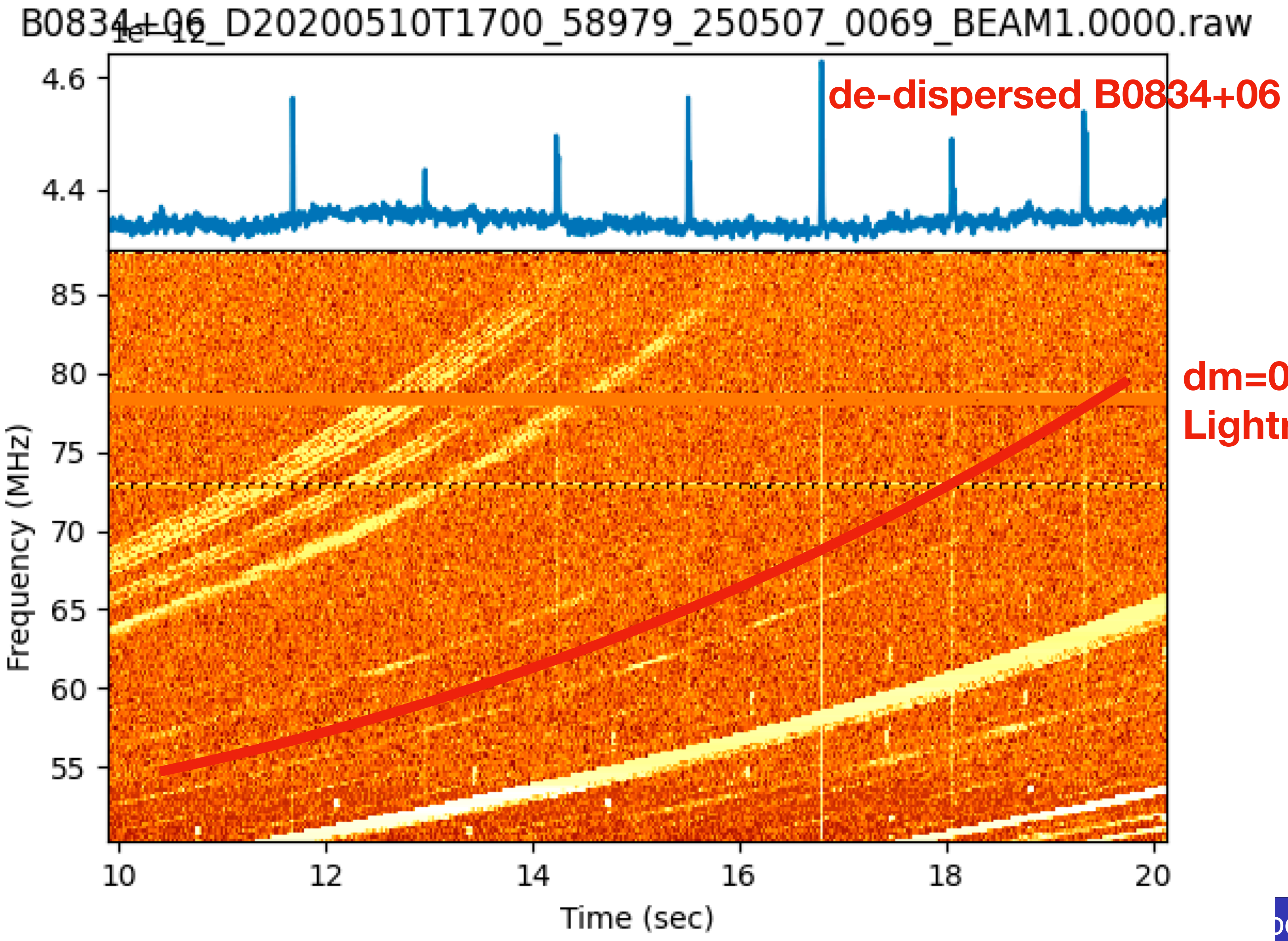
196/1 = 196 kHz **10 ms** **Total intensity**



NenuFAR waveform data with NenuRaw (dedisp dynspectrum)

Python 3 script to plot the dedispersed dynspec

```
> python3.8 /cep/lofar/pulsar/NenuRaw/demo/demo_B0834+06_dynspec.py
```



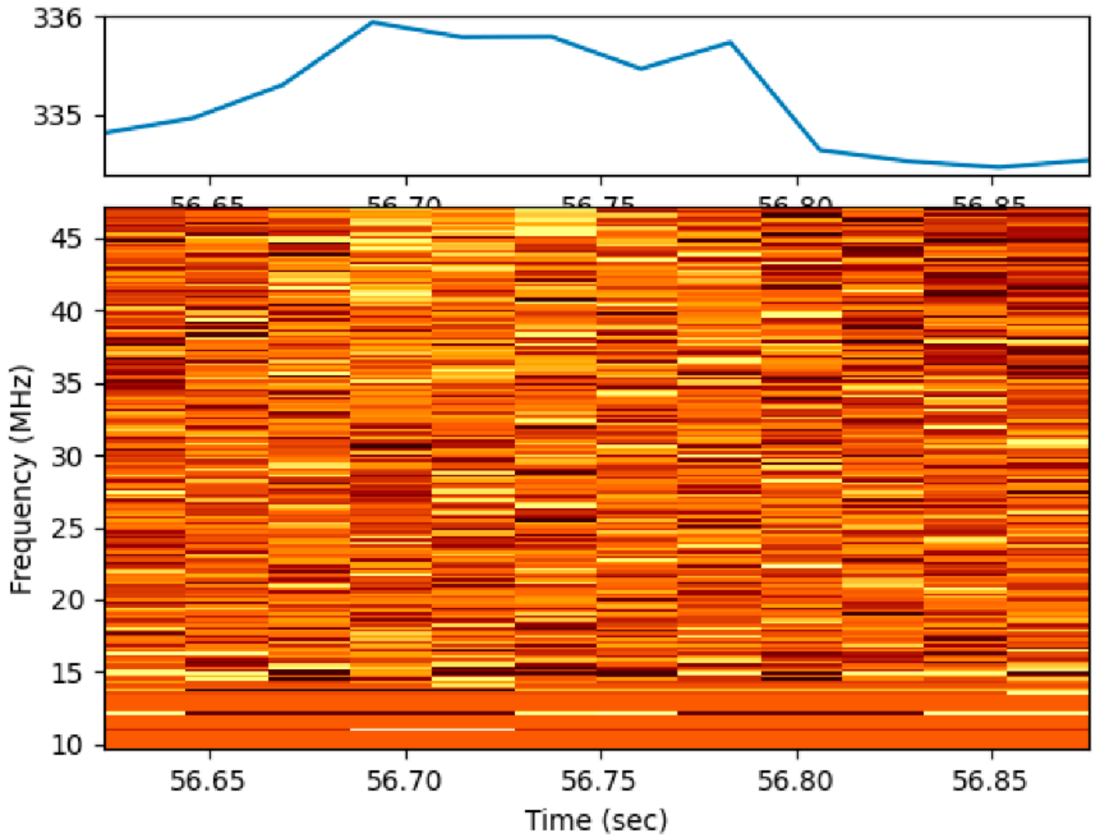
de-dispersed B0834+06

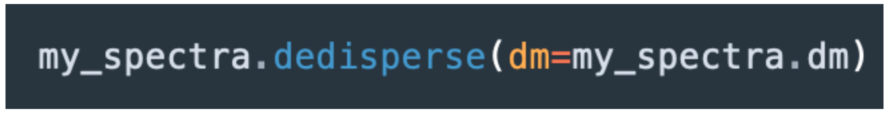
dm=0 Terrestrial Storm Lightning Pulse

NenuFAR waveform de-despersion incoherent/coherent

B0809+74 de-dispersed

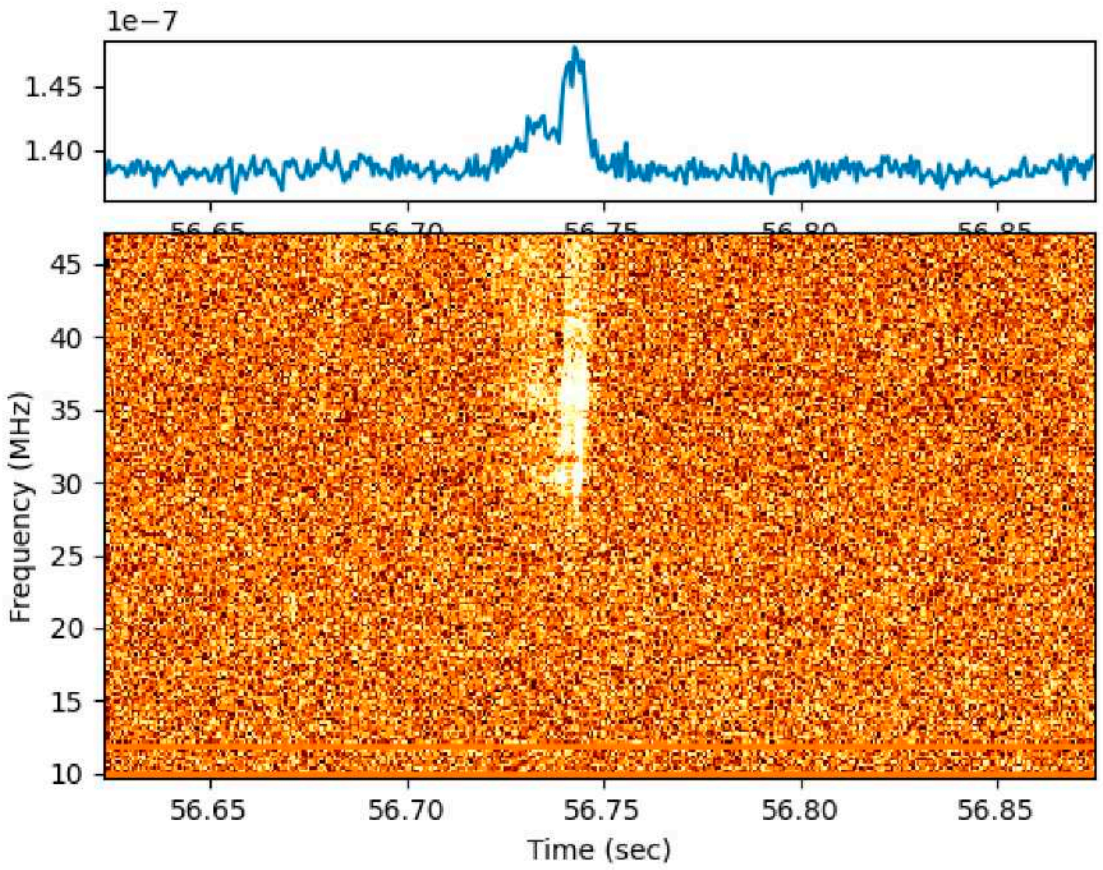
De-dispersion on total intensity

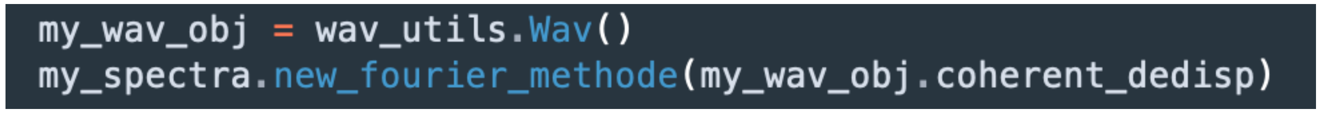




```
my_spectra.dedisperse(dm=my_spectra.dm)
```

De-dispersion on complex voltages





```
my_wav_obj = wav_utils.Wav()
my_spectra.new_fourier_methode(my_wav_obj.coherent_dedisp)
```