

Conditioning and analysis of brain signals (CABS)

Autores:

Yadira Henao

David Estepa

1 Manual de Usuario

1.1 Creación de Archivo

Antes de comenzar cualquier procesamiento de la señal se debe crear el objeto, para esto se debe ingresar la dirección .txt del archivo, y su frecuencia de muestreo. A continuación se realiza un ejemplo

```
In [9]: from Conditioning_and_analysis_of_brain_signals import Archive_EEG
        Direction='P1_RAWEEG_2018-11-15_Electrobisturí1_3min.txt'
        Archive=Archive_EEG(Direction,250)
```

El dato Archive es el objeto tipo Archive_EEG creado, el cual permitirá trabajar sobre él, y observa qué procesamiento se realiza sobre la señal.

Además de inicializar el objeto, se debe entonces también definir el tiempo de duración de cada época que se tratará en la señal. Esto se debe hacer antes de cualquier procesamiento

```
In [10]: Archive.set_Epochs(5)
```

Habiendo entonces ya definido las épocas de la señal se procede a enseñar los diferentes métodos que permiten acceder a los datos con los que se están trabajando:

```
In [11]: Archive.get_Time()
```

```
Out[11]: array([0.00000e+00, 4.00000e-03, 8.00000e-03, ..., 2.05684e+02,
                2.05688e+02, 2.05692e+02])
```

```
In [12]: Archive.get_data()
```

```
Out[12]: array([[3.043626e+04, 3.042128e+04, 3.037557e+04, ..., 3.059129e+04,
                 3.057533e+04, 3.062345e+04],
                [4.475400e+03, 4.585550e+03, 4.599250e+03, ..., 4.847690e+03,
                 4.773770e+03, 4.681150e+03],
                [1.424018e+04, 1.432510e+04, 1.432939e+04, ..., 1.477900e+04,
                 1.472077e+04, 1.465584e+04],
                ...,
                [9.200000e-02, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
                 0.000000e+00, 0.000000e+00],
```

```
[1.400000e-02, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
 0.000000e+00, 0.000000e+00],
[4.880000e-01, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
 0.000000e+00, 0.000000e+00]])
```

In [13]: Archive.get_data_processed()

```
Out[13]: array([[[[3.043626e+04, 3.042128e+04, 3.037557e+04, ..., 3.048892e+04,
 3.047097e+04, 3.042481e+04],
 [3.043881e+04, 3.047300e+04, 3.046746e+04, ..., 3.043608e+04,
 3.060861e+04, 3.046625e+04],
 [3.035910e+04, 3.043565e+04, 3.062037e+04, ..., 3.048780e+04,
 3.042236e+04, 3.047300e+04],
 ...,
 [3.067231e+04, 3.047669e+04, 3.040117e+04, ..., 3.072379e+04,
 3.051447e+04, 3.045300e+04],
 [3.051489e+04, 3.074574e+04, 3.057850e+04, ..., 3.058921e+04,
 3.055246e+04, 3.055398e+04],
 [3.060173e+04, 3.059469e+04, 3.055586e+04, ..., 3.057938e+04,
 3.062989e+04, 3.064254e+04]],
 [[4.475400e+03, 4.585550e+03, 4.599250e+03, ..., 4.498890e+03,
 4.610290e+03, 4.618560e+03],
 [4.505980e+03, 4.475180e+03, 4.582130e+03, ..., 4.535080e+03,
 4.457090e+03, 4.635460e+03],
 [4.717290e+03, 4.588320e+03, 4.476560e+03, ..., 4.648780e+03,
 4.682470e+03, 4.535120e+03],
 ...,
 [4.553230e+03, 4.739260e+03, 4.744630e+03, ..., 4.584500e+03,
 4.783000e+03, 4.789910e+03],
 [4.657750e+03, 4.576540e+03, 4.752610e+03, ..., 4.757300e+03,
 4.795100e+03, 4.689510e+03],
 [4.623660e+03, 4.725140e+03, 4.796150e+03, ..., 4.737230e+03,
 4.649160e+03, 4.724510e+03]],
 [[1.424018e+04, 1.432510e+04, 1.432939e+04, ..., 1.424837e+04,
 1.433547e+04, 1.433882e+04],
 [1.425246e+04, 1.423491e+04, 1.432157e+04, ..., 1.426844e+04,
 1.425140e+04, 1.434077e+04],
 [1.438404e+04, 1.430938e+04, 1.426971e+04, ..., 1.433232e+04,
 1.435780e+04, 1.426931e+04],
 ...,
 [1.456464e+04, 1.465119e+04, 1.464062e+04, ..., 1.459140e+04,
 1.468411e+04, 1.468628e+04],
 [1.460257e+04, 1.460867e+04, 1.468890e+04, ..., 1.468805e+04,
 1.471661e+04, 1.463375e+04],
```

[1.458849e+04, 1.466715e+04, 1.472242e+04, ..., 1.468675e+04,
1.462376e+04, 1.468326e+04]],

...,

[[9.200000e-02, 0.000000e+00, 0.000000e+00, ..., 1.840000e-01,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
...,
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 1.820000e-01, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 1.820000e-01, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00]],

[[1.400000e-02, 0.000000e+00, 0.000000e+00, ..., 3.600000e-02,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
...,
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 3.800000e-02, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 3.600000e-02, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00]],

[[4.880000e-01, 0.000000e+00, 0.000000e+00, ..., 9.700000e-01,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
...,
[0.000000e+00, 0.000000e+00, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 0.000000e+00, 9.660000e-01, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00],
[0.000000e+00, 9.620000e-01, 0.000000e+00, ..., 0.000000e+00,
0.000000e+00, 0.000000e+00]]])

In [14]: Archive.get_time_semented()

```
Out[14]: array([[0.00000e+00, 4.00000e-03, 8.00000e-03, ..., 4.98800e+00,
                4.99200e+00, 4.99600e+00],
                [5.00000e+00, 5.00400e+00, 5.00800e+00, ..., 9.98800e+00,
                9.99200e+00, 9.99600e+00],
                [1.00000e+01, 1.00040e+01, 1.00080e+01, ..., 1.49880e+01,
                1.49920e+01, 1.49960e+01],
                ...,
                [1.90000e+02, 1.90004e+02, 1.90008e+02, ..., 1.94988e+02,
                1.94992e+02, 1.94996e+02],
                [1.95000e+02, 1.95004e+02, 1.95008e+02, ..., 1.99988e+02,
                1.99992e+02, 1.99996e+02],
                [2.00000e+02, 2.00004e+02, 2.00008e+02, ..., 2.04988e+02,
                2.04992e+02, 2.04996e+02]])
```

Con los resultados anteriores se ve entonces que lo que retorna el método `get_data` será siempre la información cargada por el usuario, mientras que el método `get_data_processed` devuelve los datos procesados segmentados en épocas. Se debe tener en cuenta que dado que muchos de los métodos que se emplean eliminan épocas, la dimensión de lo que retorne este método no va coincidir con los datos originales. Para saber entonces a qué tiempo corresponde cada época que queda en los datos procesados se utiliza el método `get_time_semented`, el cual tiene la correspondencia en tiempo por cada época de cada canal.

1.2 Procesamiento de la señal

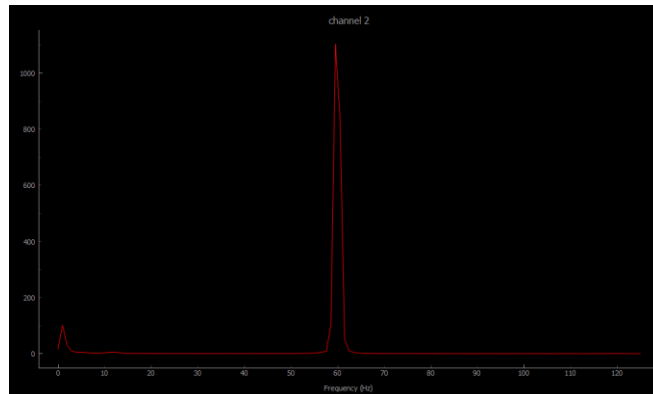
Cuando se realiza el procesamiento de la señal hay diferentes caminos a tomar, sin embargo se recomienda que el primero sea definir la banda de frecuencia en la que se desea trabajar, la cual normalmente es 1-50Hz. Para realizar esto se muestra qué función permite hacer esto.

In [16]: Archive.Prefilter();

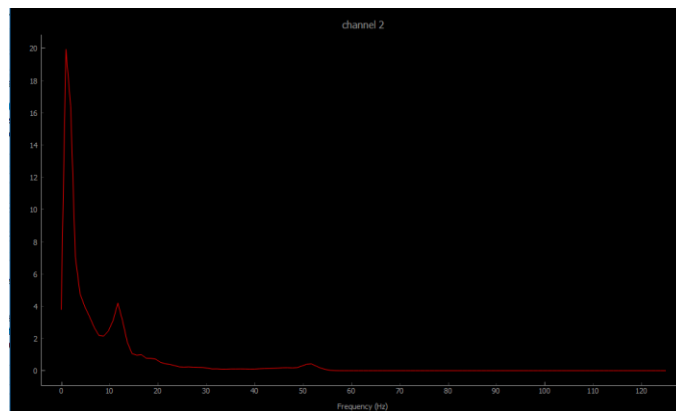
El filtrado que se realizó fue sobre los datos procesados. Para mostrar que realmente se está trabajando sobre la banda de frecuencia escogida se recurre entonces enseñar los métodos que permiten realizar los gráficos del espectro de potencia de la señal.

El método `plot_Spectrum(Channel,Procesadas)` presenta los dos argumentos que se muestran, siendo `Channel` para decidir si realizar el espectro de un solo canal, o si de todos (No ingresando un valor de canal), y el argumento `Procesadas` establece si se realiza el espectro de la señal procesada o si no. A continuación se muestran ejemplos.

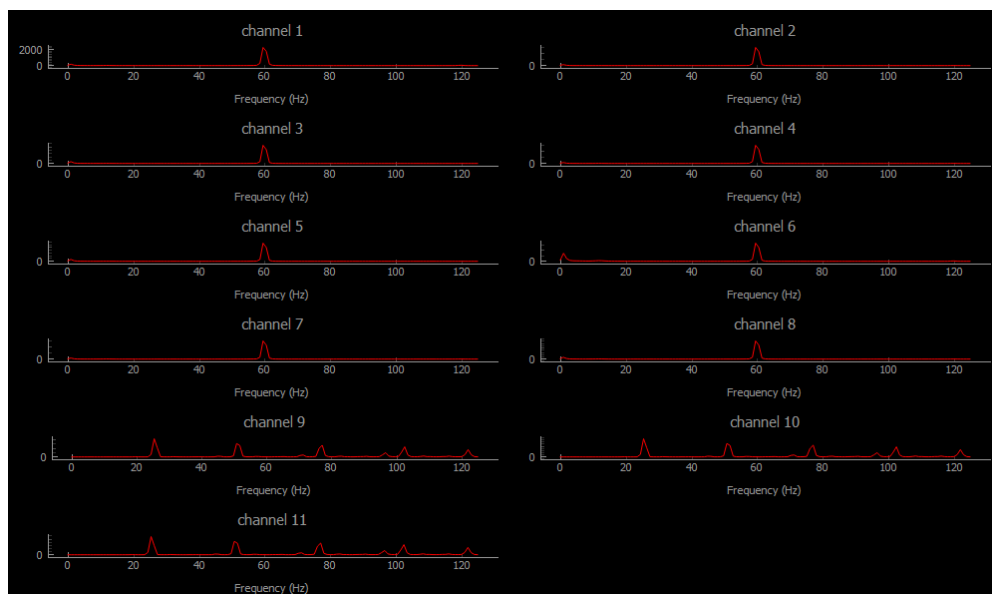
In [17]: `Archive.plot_Spectrum(Channel=2)`



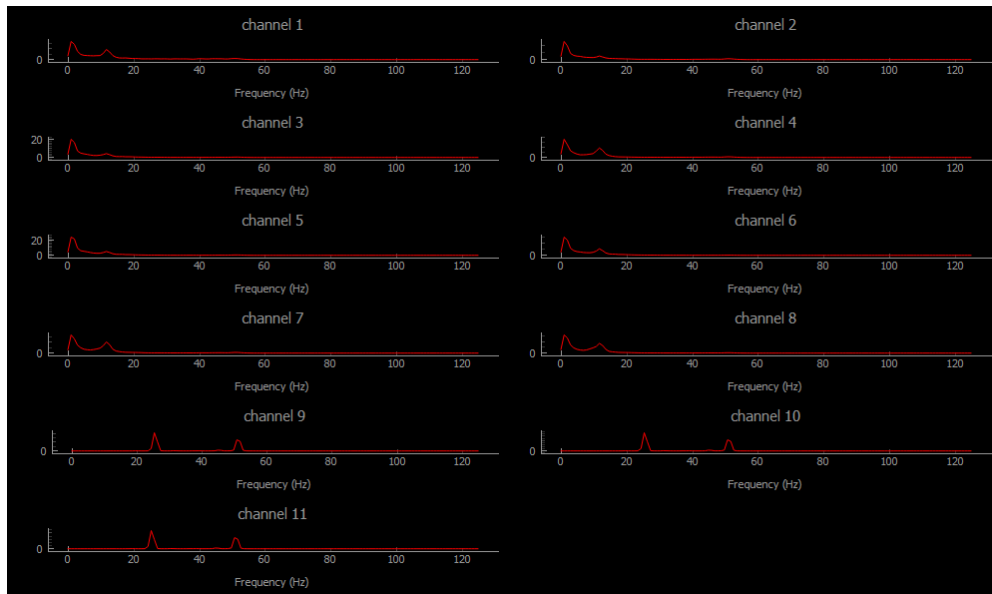
In [18]: `Archive.plot_Spectrum(Channel=2,Procesadas=True)`



In [19]: `Archive.plot_Spectrum()`



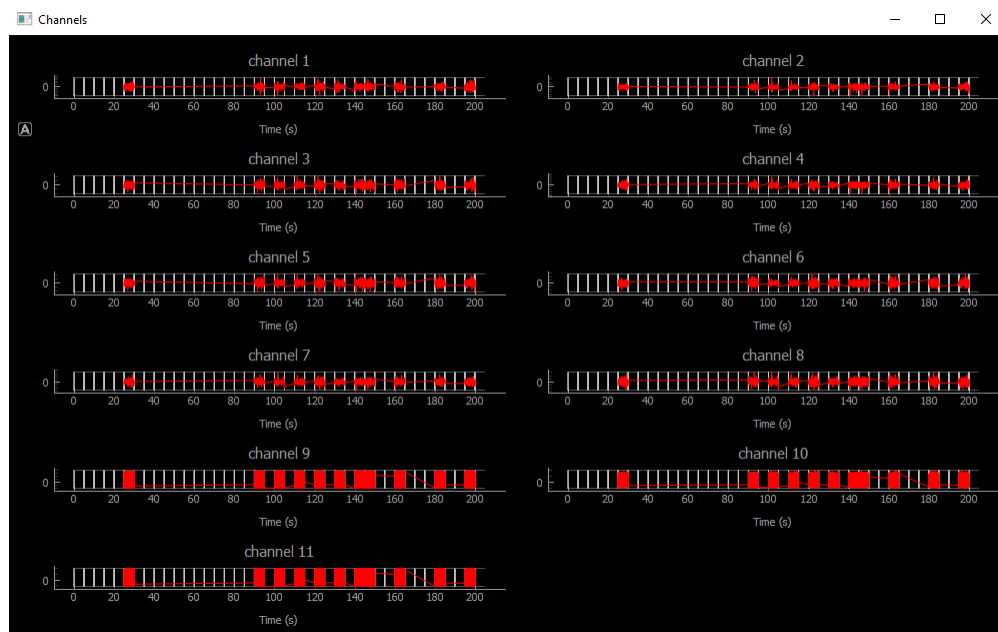
In [20]: `Archive.plot_Spectrum(Procesadas=True)`



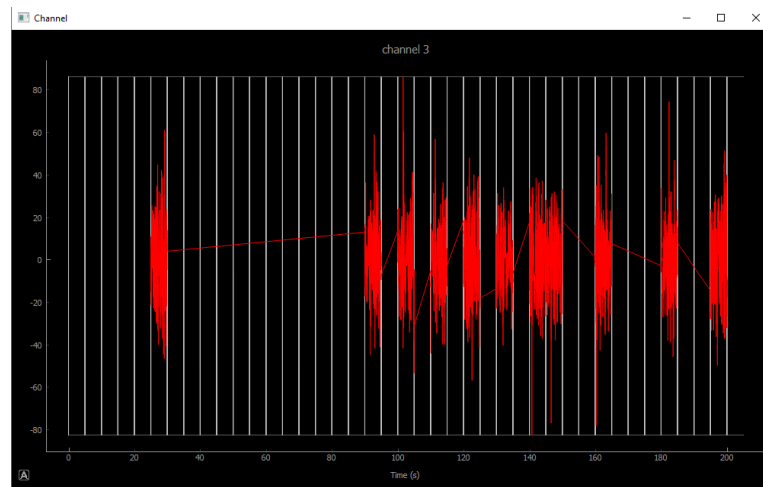
Como se observa en las gráficas que se generan, cada vez que se grafica las procesadas se encuentra que su rango de frecuencia esta entre 0-50Hz, como se había realizado con anterioridad.

Si se deseara ver la señal temporalmente se realiza mediante el método `plot(Segmentacion,Procesadas,Channel)` donde los argumentos permiten: `Segmentacion` decide si se muestra una cuadrícula que delimita las épocas, `Procesadas` define si se grafican los datos procesados o no, y `Channel` permite decidir qué canal graficar, o si no se define, grafica todos.

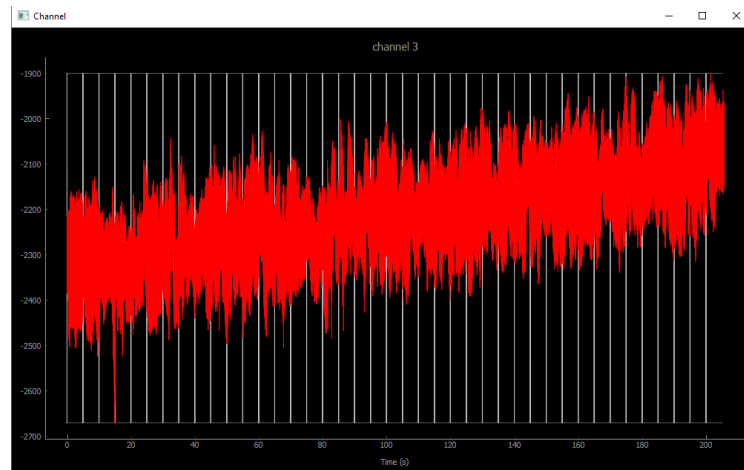
In [21]: `Archive.plot(Segmentacion=True,Procesadas=True)`



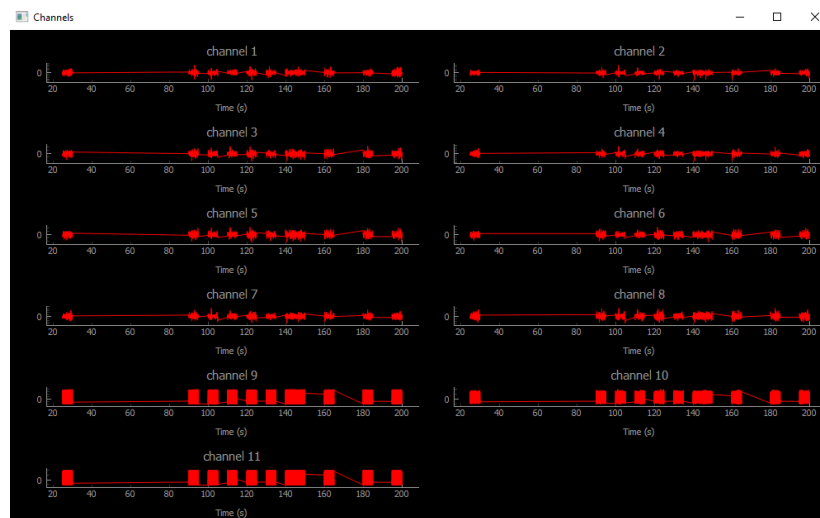
In [22]: Archive.plot(Segmentation=**True**,Procesadas=**True**,Channel=3)



In [23]: Archive.plot(Segmentation=**True**,Channel=3)



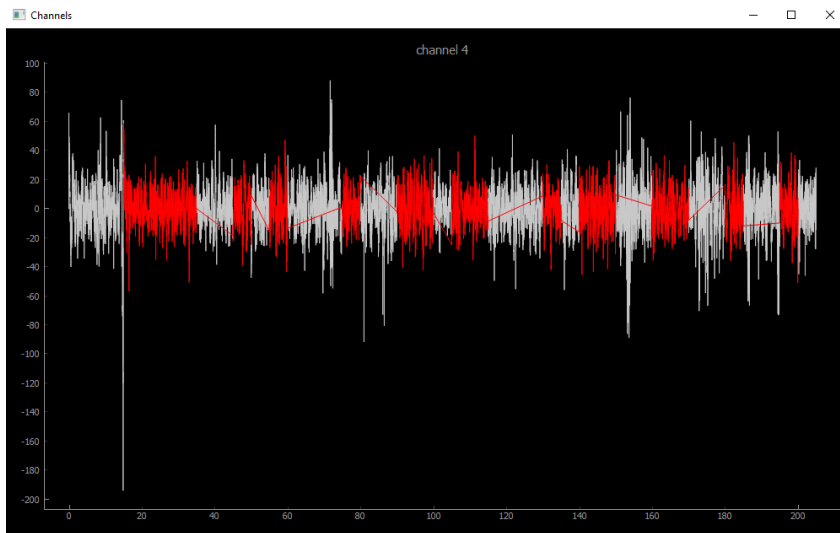
In [24]: Archive.plot(Procesadas=**True**)



Luego de mostrar los métodos generales de graficación, se mostrarán los diferentes métodos de rechazo con los que cuenta la librería, mostrando a su vez otro método de graficación que permite realizar comparación entre diferentes métodos empleados.

In [25]: `Archive.Limit_value(100)`

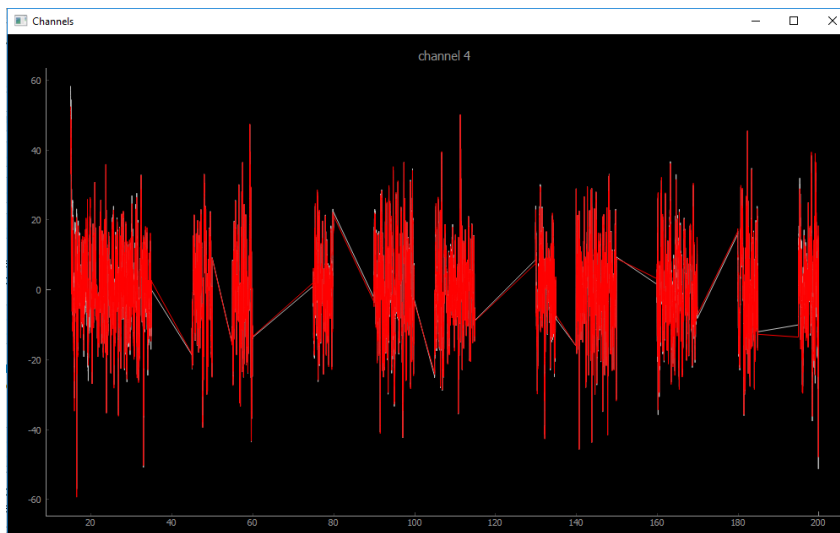
`Archive.plot_Vs(4)`



El método utilizado anteriormente realiza un rechazo de época por valor umbral, este umbral es ingresado por el usuario. El método `plot_Vs(Channel)` permite graficar los cambios que se realizaron en la señal procesada luego de realizar el método de rechazo, siendo la señal blanca los datos eliminados por el método.

In [26]: `Archive.remove_linear_tend()`

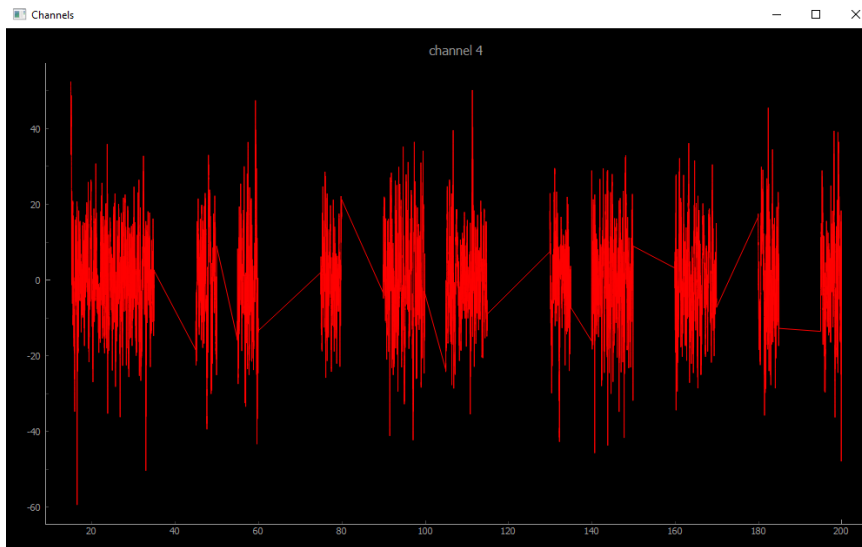
`Archive.plot_Vs(4)`



En este caso se emplea el método de eliminación de tendencia lineal, el cual en este caso no sirve de mucho ya que al aplicar el filtro de 1-50 Hz se eliminaron componentes DC, entre esas las posibles tendencias lineales que tuviera la señal.

In [27]: `Archive.Kurtosis_method(-3,2)`

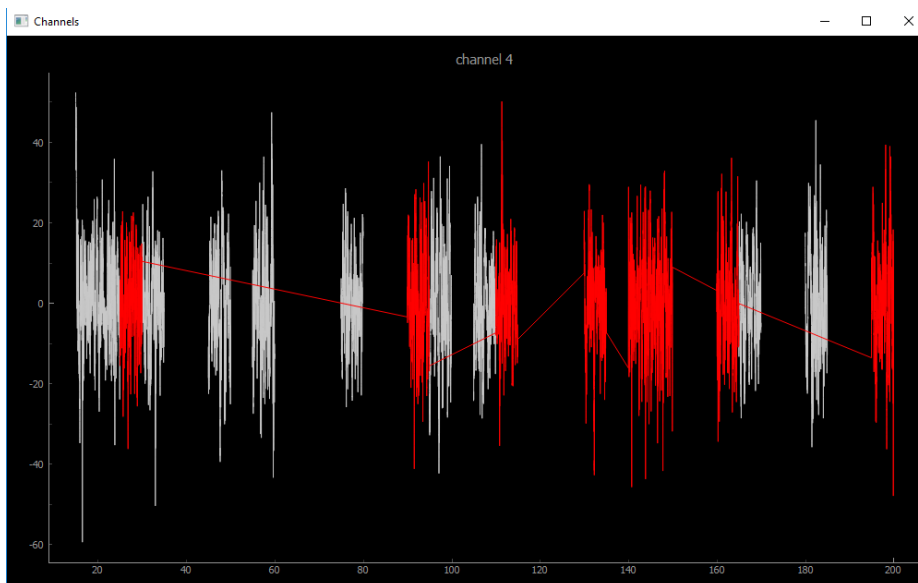
`Archive.plot_Vs(4)`



En este caso se emplea el método `Kurtosis_method(valor_mínimo,valor_máximo)` el cual elimina datos que tengan distribuciones improbables en señales EEG. Este calcula la curtosis de una época, y si el valor dado se encuentra por fuera del rango ingresado por el usuario, la época es removida. En este caso se encuentra que no hubieron épocas removidas.

In [29]: `senal_final=Archive.Spectral_method(60)`

`Archive.plot_Vs(4)`

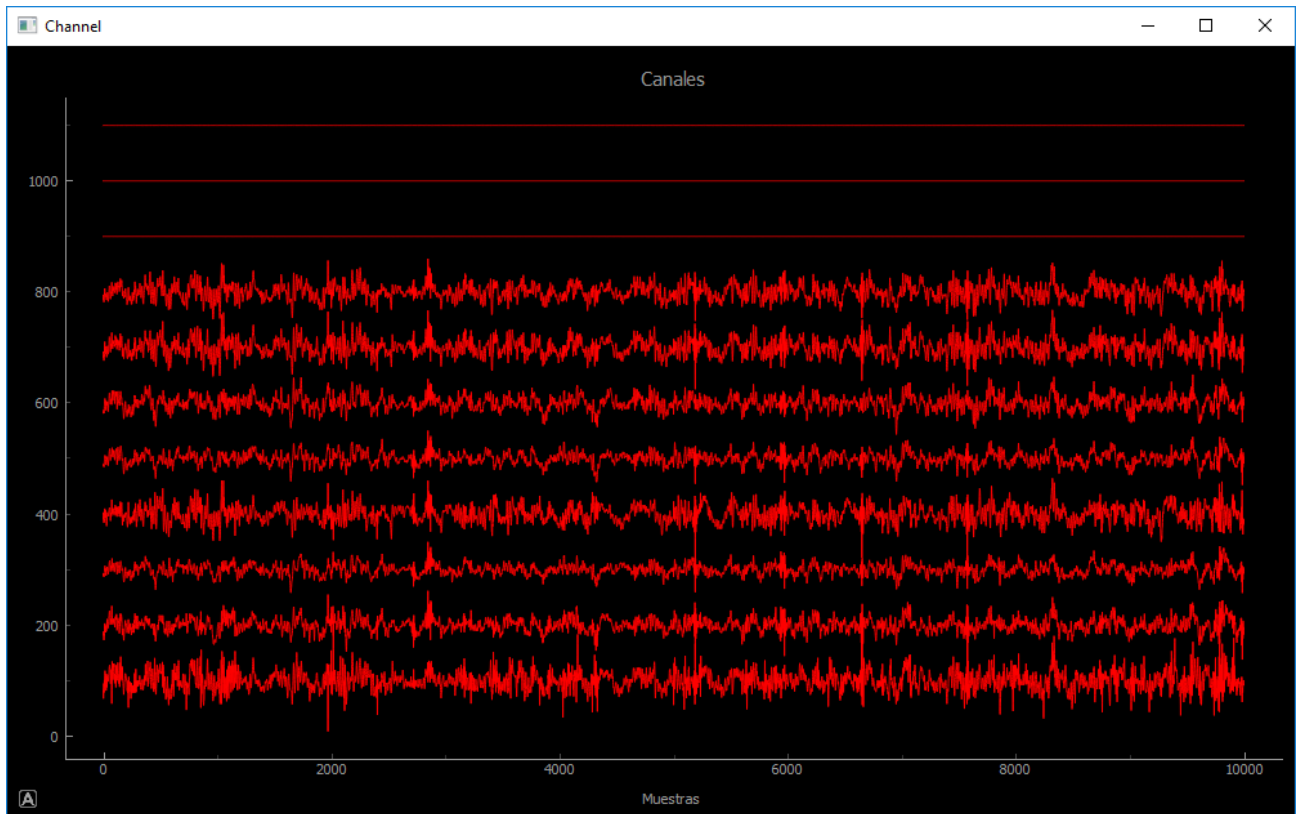


El método `spectral_method(Umbral)` elimina datos que tengan componentes frecuenciales no características del EEG, para esto se compara el espectro de cada época con el espectro general del canal, y si de la resta entre el promedio de la potencia del espectro del canal y el espectro de la época, se encuentra un valor por encima del umbral establecido, la época es removida.

Para finalizar se mostrará un código que permite graficar las señales resultantes sin la discontinuidad generada por eliminar las épocas. Se debe tener en cuenta también que cada método de remoción de época retorna los datos procesados, por lo que un análisis externo de estos es posible realizar.

```
In [31]: from pyqtgraph.Qt import QtGui,QtCore import
pyqtgraph as pg import sys import numpy as
np x,y,z=senal_final.shape
senal=senal_final.reshape((x,y*z))

win = pg.GraphicsWindow(title="Channels EEG")
win.resize(1000,600) win.setWindowTitle('Channel')
pg.setConfigOptions(antialias=True)
P=win.addPlot(title='Canales') for canal in np.arange(x):
    P.plot(senal[canal]+100*(canal+1),pen=(255,0,0)) P.setLabel('bottom', 'Muestras') if
(sys.flags.interactive != 1) or not hasattr(QtCore, 'PYQT_VERSION'):
    QtGui.QApplication.instance().exec_()
```



En la gráfica anterior se tiene cada canal graficado de abajo hacia arriba, siendo la primera gráfica el canal 1, y la última el canal 11.