Python Processing Infrasound Through an mSeed File

Souleymane Cissokho - Department of Mechanical Engineering - Rose-Hulman Institute of Technology – Abstract #149

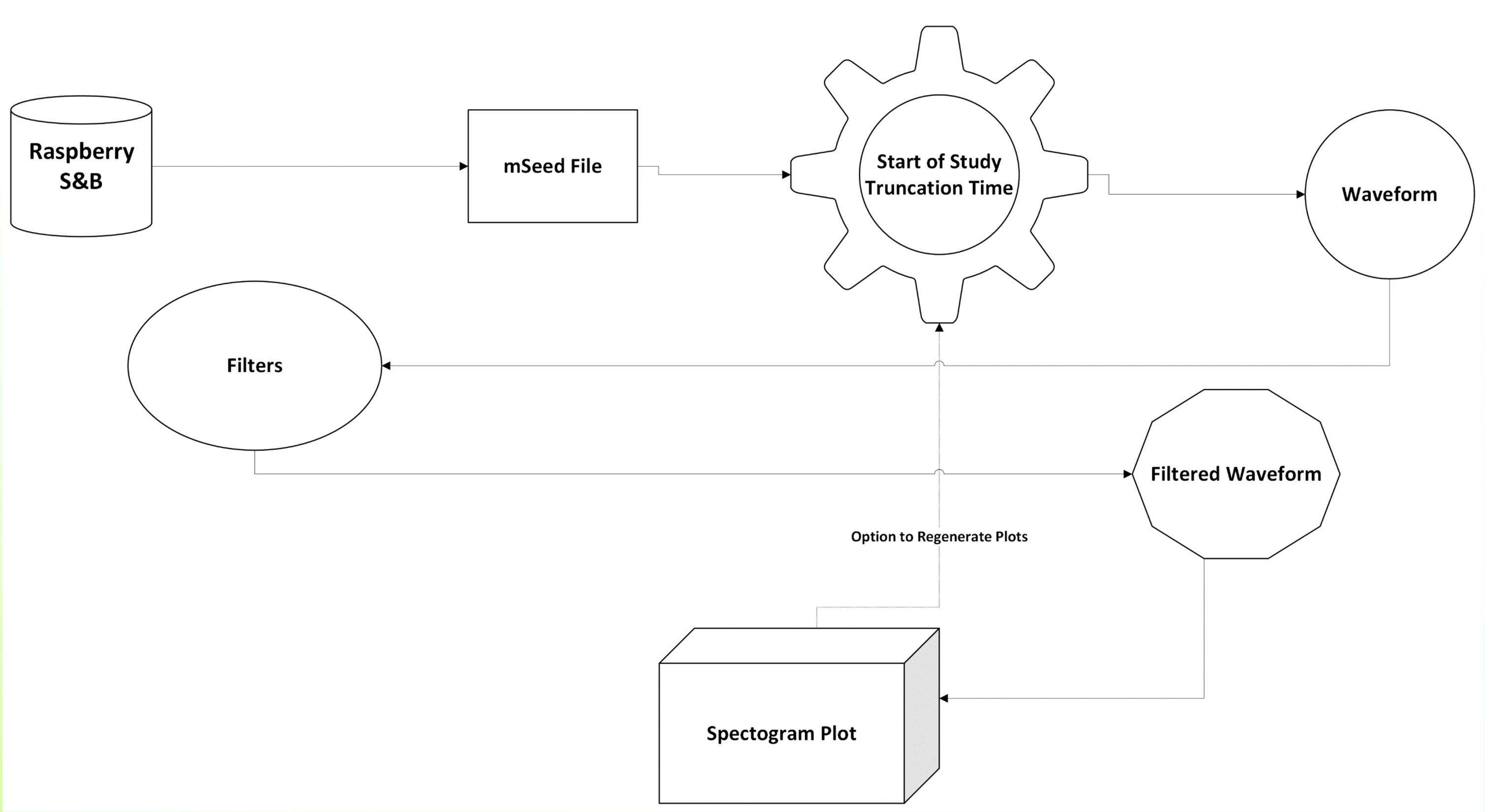
Luis Ocampo Giraldo - Radiochemistry & Nuclear Measurements – Idaho National Laboratory

Background

- Infrasound are low sound frequencies that humans cannot hear
- Can be recorded through commercial-off-the-shelf sensors
- Data can be used to characterize explosions and detect soil landslides [1][2]

Data Processing Method

- Data taken from sensor, runs through a regenerative Python function that performs rapid infrasound analysis
- Data must be in an mSeed file format



Data Exportation

- Raspberry Shake & Boom Seismoacoustics sensor
- Recorded infrasound data over 24-hours with 100 Hz sample rate
- 8.6 million samples taken during this time
- Exported data as an mSeed seismic file

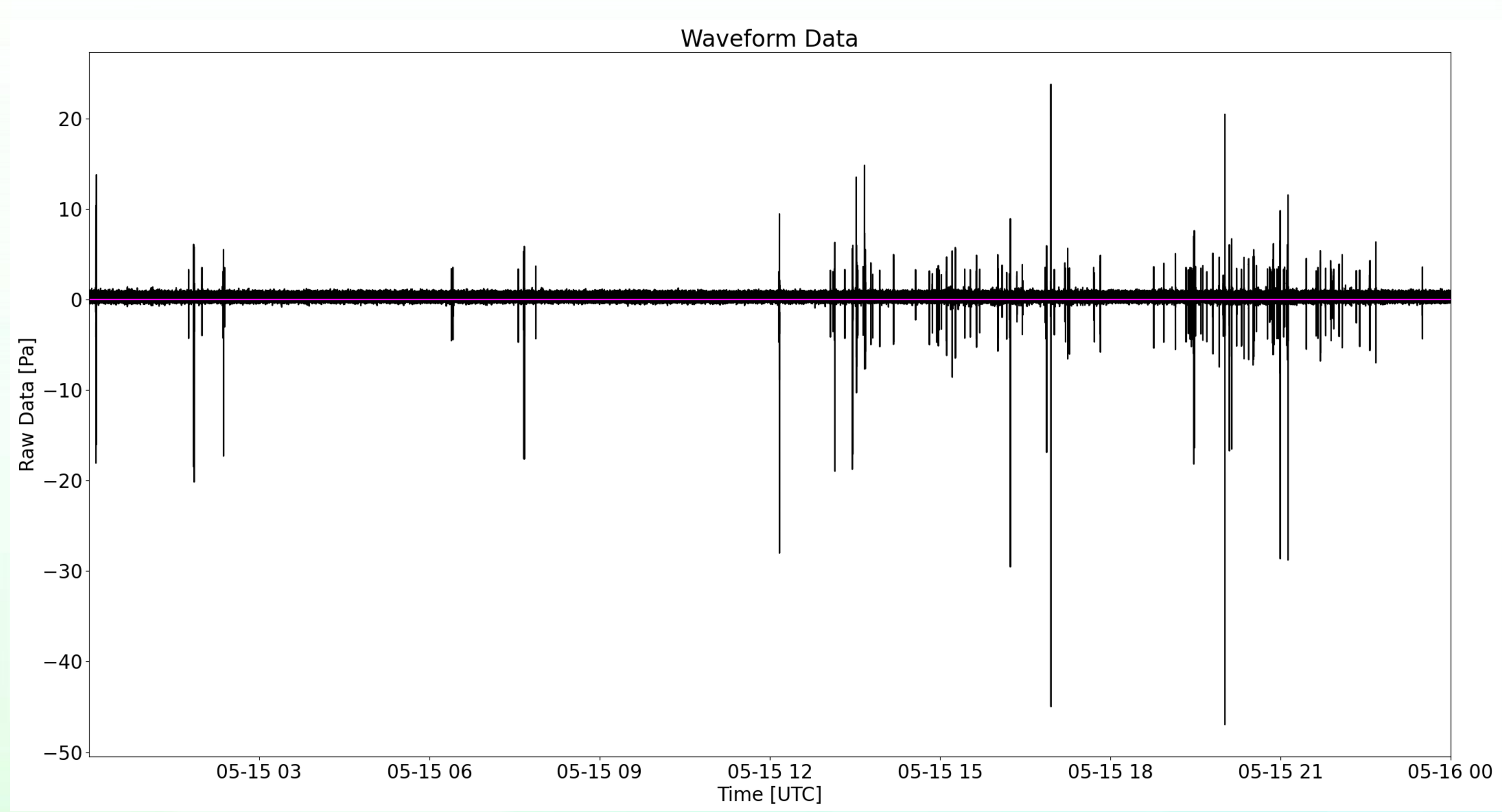
Python Function

- Python function created to process data
- Libraries used include ObsPy, Numpy, Matplotlib, and Pandas
- Function contains 3 inputs:
  - Path to data file (String)
  - Start of study (String in UTC format)
  - Truncation time (int, minutes)

```python
from obspy import read, UTCDateTime
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
def interactiveFeatureAnalysis(dataFile : str,truncationMin, startingDateTime):
    """This function has to be used with an external script to input the datafile (mSeed),
    truncation time in minutes, and the starting time of study in UTC format
    - utilizes obspy, numpy, matplotlib, and pandas
    Args:
        dataFile (mSeed -  string): the string must be a path to the mSeed file to be analyzed
        truncationMin (min - int): truncation time in minutes
        startingTime (UTC, string): UTC time (up through min or seconds) of the file that you want to start analyzing -
       "2024-05-15T07:00" is the proper format
    """
```

```python
def interactiveFeatureAnalysis(dataFile : str,truncationMin, startingDateTime):

    datasetFiltered.filter('highpass', freq = 0.1) # puts data through a high pass filter
    datasetFilteredSR.filter('highpass', freq = 0.1) # puts data through a high pass filter

    ultBeginningTime = np.datetime64(tr.stats.starttime.__str__())
    ultEndTime = np.datetime64(tr.stats.endtime.__str__())
    dtUTCForm = UTCDateTime(startingDateTime)
    dtDateTimeForm = np.datetime64(startingDateTime)
    endTimeDateTimeForm = dtDateTimeForm + np.timedelta64(truncationMin, 'm')
    utcRangeRaw = pd.date_range(start=ultBeginningTime, end=ultEndTime, periods=len(tr))

    dataset500 = datasetFiltered.copy()
    dataset500SR = datasetFilteredSR.copy()

    dataset500.trim(dtUTCForm,dtUTCForm + (60*truncationMin) - (1/100))  # the minus 1/100 removes the last point due to the trimming adding an extra value
    dataset500SR.trim(dtUTCForm,dtUTCForm + (60*truncationMin) - (1/100))  # the minus 1/100 removes the last point due to the trimming adding an extra value

    dataset499 = datasetLowFilter.copy()
    dataset499SR = datasetLowFilterSR.copy()

    dataset499.trim(dtUTCForm,dtUTCForm + (60*truncationMin) - (1/100))  # the minus 1/100 removes the last point due to the trimming adding an extra value
    dataset499SR.trim(dtUTCForm,dtUTCForm + (60*truncationMin) - (1/100))  # the minus 1/100 removes the last point due to the trimming adding an extra value

    # utcRangeFiltered = pd.date_range(start=ultBeginningTime, end=ultEndTime, periods=len(datasetLowFilter))
    utcRange500 = pd.date_range(start=dtDateTimeForm, end=endTimeDateTimeForm, periods=len(dataset500))


    # plotting just the original data
    plt.plot(utcRangeRaw, datasetLowFilter,'k')
    plt.ylabel('Raw Data [Pa]').set_fontsize(20)
    plt.xlabel('Time [UTC]').set_fontsize(20)
    plt.tick_params(axis='both', which='major', labelsize=20)
    plt.ticklabel_format(axis='y', style='plain')
    plt.xlim(utcRangeRaw[0], utcRangeRaw[-1])
    plt.axhline(y=0, color = 'magenta', linestyle = '-') # Horizontal magenta line to signify dc offset
    plt.ticklabel_format(axis='y', style='plain')
    plt.title('Waveform Data').set_fontsize(24)
    plt.show()
```

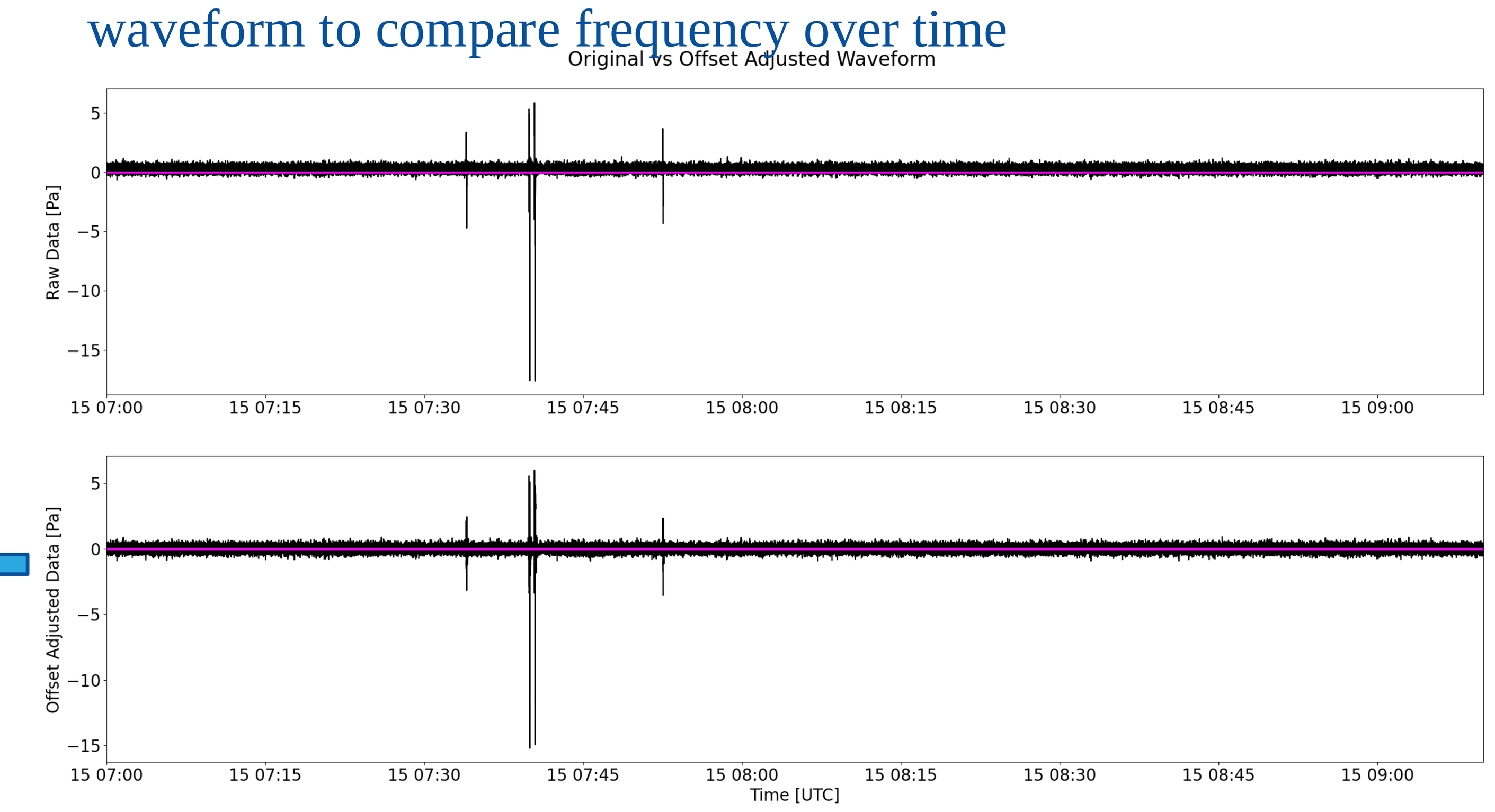
- Function sorts, trims, and filters data based of what the user chooses

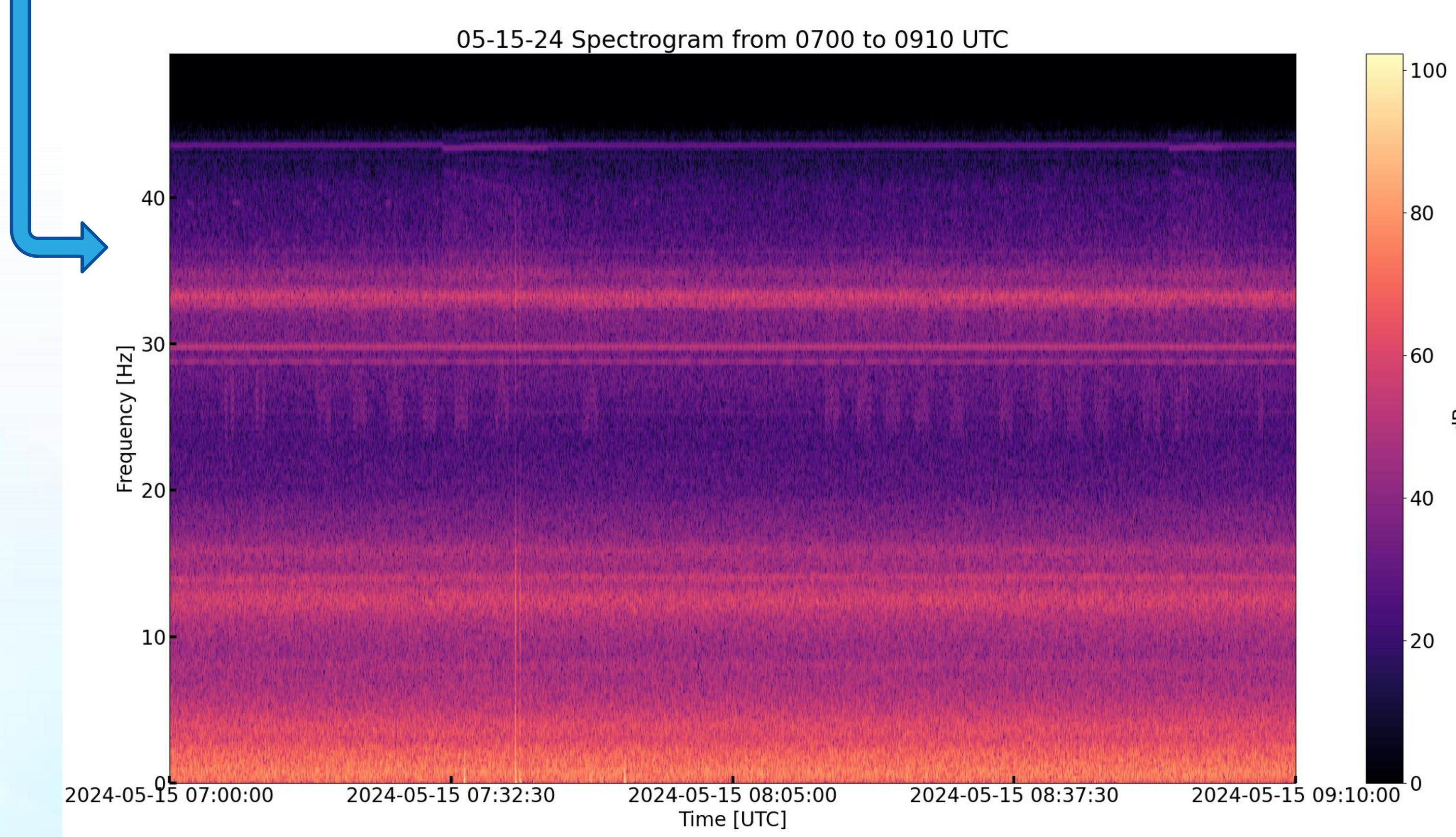


- Waveform for entire data set, with time in UTC on horizontal, and amplitude in pascals on the vertical

Filtering & STFT

- Waveform truncated based off argued start of study and truncation time
- Waveform is not normalized at 0 Pa
- High-pass filter is used to remove DC offset
- Short-Term Fourier Transform (STFT) is performed on filtered waveform to compare frequency over time





- Function produces spectrogram from the offset adjusted data
- Python function is also interactable with any script and with any other sensor that produces an mSeed file

References

[1] A. Cannata, P. Montalto, E. Privitera, G. Russo, and S. Gresta, "Tracking eruptive phenomena by infrasound: May 13, 2008 eruption at Mt. Etna," Geophysical Research Letters, vol. 36, no. 5, Mar. 2009, doi: https://doi.org/10.1029/2008gl036738.

[2] D. Liu, D. Tang, S. Zhang, X. Leng, K. Hu, and L. He, "Method for feature analysis and intelligent recognition of infrasound signals of soil landslides," Bulletin of engineering geology and the environment, vol. 80, no. 2, pp. 917–932, Oct. 2020, doi: https://doi.org/10.1007/s10064-020-01982-w.

[3] L. A. Ocampo Giraldo, E. S. Cardenas, M. N. Patterson, "Acoustic Monitoring of Pyroprocessing Equipment," Nuclear Energy Agency 16th Information Exchange Meeting on Actinide and Fission Product Partitioning and Transmutation (16IEMPT) Proceedings, (2023) INL/CON-23-73773.

[4] Raspberry Shake and Boom Sensor. Accessed: Jun. 04, 2024. [Online]. Available: https://shop.raspberryshake.org/product/turnkey-iot-atmospheric-infrasound-monitor-rsboom/?attribute_pa_variation=indoor&attribute_pa_license=private-use-125-discount

Acknowledgements

This research was funded through a Laboratory Directed Research and Development project under Battelle Energy Alliance, LLC contract number DE-AC07-05ID14517. I would like to thank my mentor, Dr. Ocampo Giraldo for his guidance on this project. I would also like to acknowledge the advice given by Dr.Applbaum, Dr. Hart, and Dr. Watrous and for their help to set me up for success at INL.

www.inl.gov

INL/CON-24-80035

THE NATIONAL GEM CONSORTIUM

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Battelle Energy Alliance manages INL for the U.S. Department of Energy's Office of Nuclear Energy

Idaho National Laboratory