LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	LIGO-T1900287–v1	2020/01/22

Data Clustering Techniques for the Correlation of Environmental Noise to Signals in LIGO Detectors

Jacob Bernhardt, Anamaria Effler, Rana Adhikari

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW22-295
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

1 Introduction

The LIGO project uses laser interferometry to measure gravitational waves (GWs). LIGO interferometers transduce their relative arm length differences caused by GWs to a signal composed of optical power, known as DARM.

Due to the amplitude scales of GWs, The LIGO detectors have to operate at a very high sensitivity; the spectral density of a measurable length difference is as low as 2×10^{-20} m/$\sqrt{\text{Hz}}$ at 100 Hz. The design of earthbound LIGO is thus heavily focused on the filtering and isolation of environmental noise.

To help identify and characterize environment-based noise, the LIGO detector has a Physical Environment Monitoring (PEM) system, a diverse array of environmental sensors positioned all over the facility[1]. This is used for a multitude of purposes, including the data quality report (DQR), which aims to veto segments of time by finding correlations between PEM and DARM through statistical inference and sometimes also supervised learning. Supplementing coincidence analysis between the two detectors, DQR prevents GW-like noise transients from being falsely categorized as events.

Both detector livetime and detection range can be increased by figuring out how to decouple environmental noise from DARM. Directly coupling noise, found by basic coherence, has been already addressed, but the complexity of the detector causes many noise sources to up- or down-convert. These require some more careful statistical correlation to identify, and are sometimes not well understood.

Separating noise sources out of a signal can be considered a clustering problem in a space covering different frequency bands in which noise appears. A previous LIGO SURF student has evaluated several data clustering algorithms with respect to their ability to properly sort out frequency elements of seismometer signals caused by specific earthquake events[2]. Both the k-means algorithm, which aims to make clusters with low standard deviation, and the DBSCAN algorithm, which minimizes overall inter-point distance in clusters, were evaluated using multiple methods, including the Calinsky-Harabaz index and direct comparison to earthquake times via time labeling of points, ultimately showing poor earthquake identification. A long short-term memory (LSTM) recurrent neural network (RNN) seemed to work much better, but due to small input sample size, this solution may have been be plagued by over-fitting. Thus, it is imperative that a more robust frequency clustering mechanism be designed for the PEM system.

2 Objectives

1. Find an algorithm or clustering approach which correctly identifies known noise events.
2. Use the results of (1) to create a clustering approach to discover previously unknown noise correlations and/or sources.

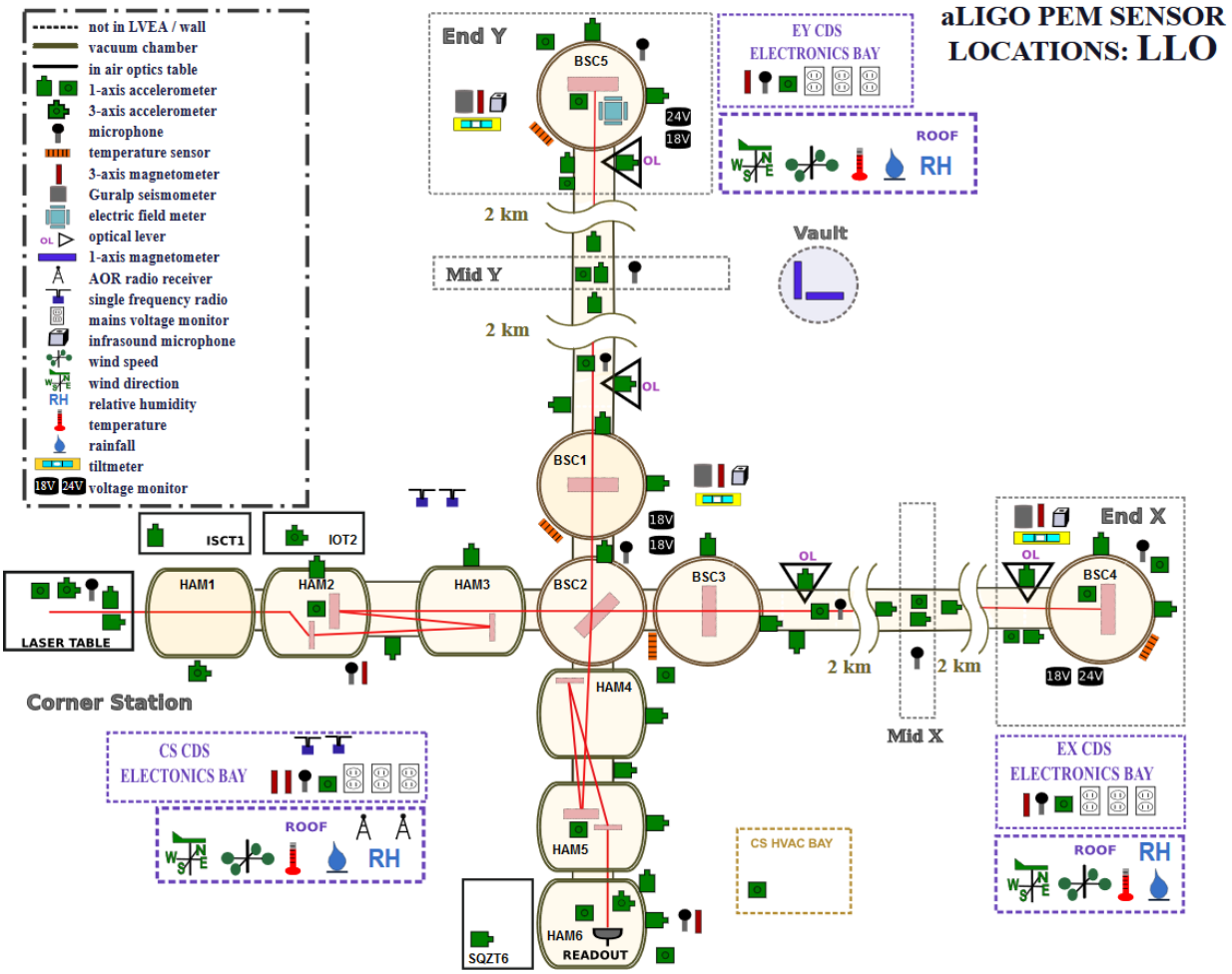

Figure 1: Schematic PEM map at the LIGO Livingston Observatory (L1). Shaded areas are in vacuum.

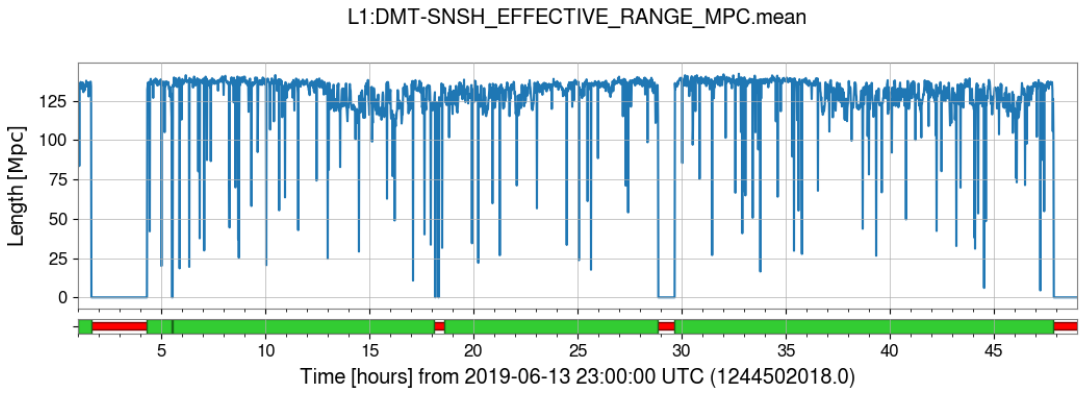

Figure 2: Detector range at L1 seems to consistently reduce during the day (∼6am-5pm CST), thought to result from anthropic activity. This can be as much as 10 Mpc for NS-NS mergers.

3 Approach

A program was written to take the spectral power of any PEM channel, in the form of band-limited RMS (BLRMS) by summing in-band bins of a power spectrogram, with inspiration from [3]. By clustering spectral bands of PEM channels, frequency conversions could be accounted for.

A Python script was written to perform the an initial clustering approach, explained in Section 4, with a variety of tunable parameters. It was validated on a regime with known noise states, the seismometers, satisfying the first objective. No other clustering approach was extensively investigated (Appendix B).

Unknown noise states in accelerometers and microphones were clustered using this approach, to meet the second objective. (Section 5).

To glean the correlations between frequencies in clustered channels, a script was created to visually present the power in frequencies in channels by cluster. This is explained in Section 6.2.

4 k-Means Clustering with Histories

The k-means algorithm was used to cluster the two hours of minute-trend data preceding each point in time. The coordinates of a clustered point were as follows:

$$\begin{gathered}\{s_0(t_0), s_0(t_{-1}), s_0(t_{-2}), \cdots, s_0(t_{-n}),\\ s_1(t_0), s_1(t_{-1}), s_1(t_{-2}), \cdots, s_1(t_{-n}),\\ s_2(t_0), s_2(t_{-1}), s_2(t_{-2}), \cdots, s_2(t_{-n}),\\ \cdots,\\ s_m(t_0), s_m(t_{-1}), s_m(t_{-2}), \cdots, s_m(t_{-n})\}\end{gathered} \tag{1}$$

with $s_j(t)$ the value of a clustered channel j at time t in its own units, e.g. a seismometer velocity at time t. Each dimension could be thought of as "value of specific channel a specific number of minutes ago", allowing trends over time to be matched together in a phase-agnostic way. This yields a cluster space of dimensionality (# of channels) $\times$ (# of minutes of history).

Notably, the clustering endeavored by [2] lacked this history feature, using a clustering space of dimensionality (# of channels) $\times$ (1). Instead of just finding times when the channels are similar in value, the history clustering is sensitive to the shape of clustered features.

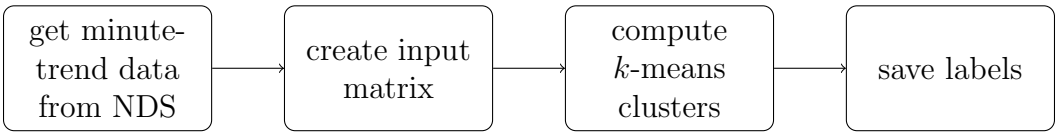

Figure 3: Clustering script flowchart. The clustering algorithm creates a time series of cluster numbers, a label for each minute of clustered time.

4.1 Sanity Checking with Seismometers

For a total clustering duration of 30 days, using the seismometers attached to ETMY, ETMX, and ITMY, in minute-trend half-order-of-magnitude BLRMS bands from 30 mHz to 30 Hz, the following known noise events were easily identified using a "2-hour history" k-means method:

- earthquakes (0.01 → 0.1 Hz)
- microseism (0.1 → 1 Hz)
- anthropogenic noise (1 → 10 Hz)

A microseism is a faint seismic tremor caused by natural processes, such as waves of pressure in the ocean, which happens over the course of hours. All human activity creates so-called "anthropogenic noise", and cars, trains, logging, and even footsteps can couple to DARM. The noise floor at anthropogenic-noise-dominated frequencies noticeably rises during the day and lowers during the night. Loud human-originated events, such as trains passing, stand out over even the daytime noise floor.

Some differentiation between subcategories of events in the same frequency band but of different timescales (e.g. earthquakes vs. wind; train vs. noise from cars) was lacking.

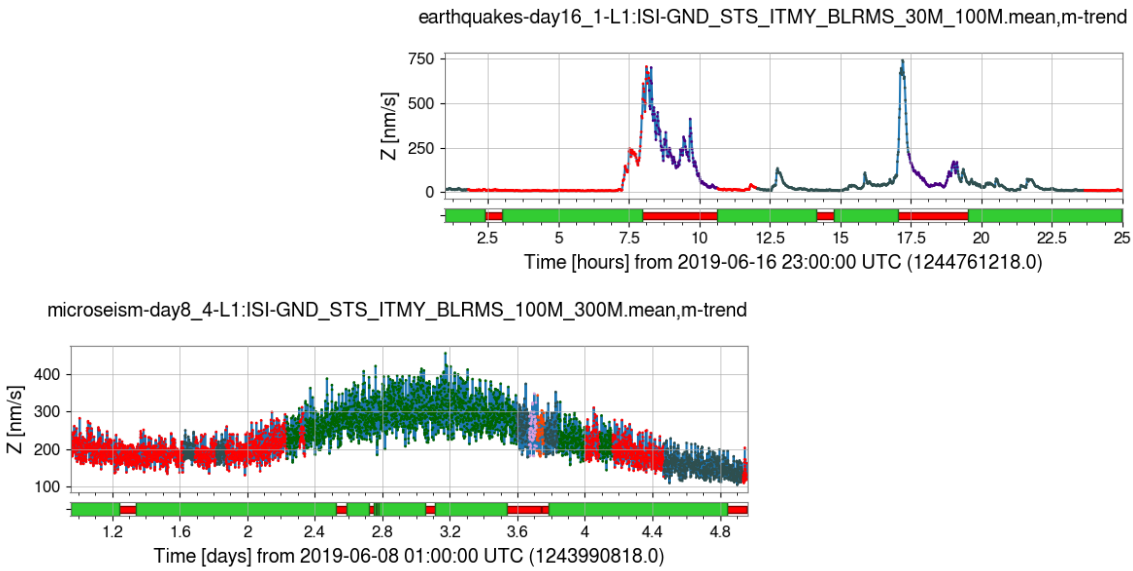

Figure 4: Seismic BLRMS near clustered earthquakes (above, purple, 30 mHz → 0.1 Hz) and microseism (below, green, 0.1 Hz → 0.3 Hz)

The length of the history was initially thought to have an effect on the timescales of identifiable events; experimentation (namely, trying 30-minute and 6-hour histories on the same data) showed that this is not really true.

The next idea was that there were too many different types of features in too large a space for events with a small number of points, like the trains, to be separated out. The test was

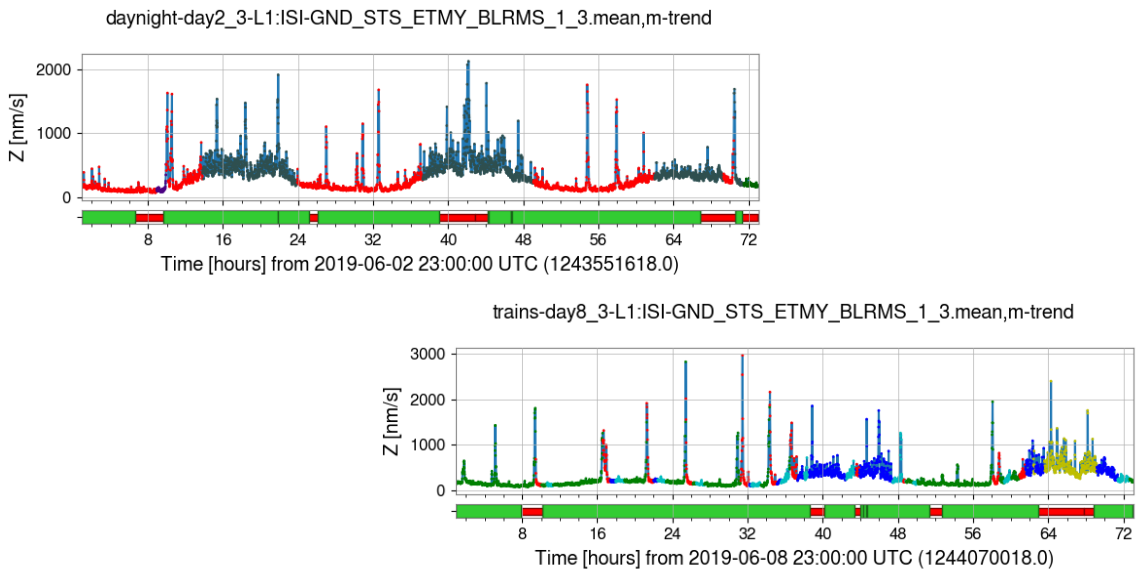

Figure 5: Seismic BLRMS from 1 Hz $\rightarrow$ 3 Hz. Using all bands, the k-means approach clustered all anthropogenic features together (above, gray). Limiting the clustering space allowed short events like trains (red, below) to be separated out from day/night variation (blue, below).

re-run with only anthropogenic seismic BLRMS bands at the end stations, which yielded very clear distinction between the anthropogenic noise types (See Figure 5).

5 Clustering of Unknown Noise States

Now that the clustering scheme was verified using known states of noise, it could be used to find new relationships between PEM channels and DARM noise.

5.1 Microphones

A "2-hour history" k-means method was used to cluster 30 days of data from microphones in the LVEA, XVEA and YVEA in BLRMS bands shown in Table 1. Two interesting "noisy" states were identified: a relatively quiet, multi-hour cluster, and a quick, loud cluster. Tables of the median percent increase of amplitude in certain BLRMS bands with respect to the largest cluster in the run (representative of the noise floor) are shown in Table 2, and their spectra in Figure 6.

The clustering was repeated with observing segments of `L1:GDS-CALIB_STRAIN` included. This produced approximately the same clusters as before, likely because segmenting of DARM added weight to the MIC channels. The cluster corresponding to the same longer, quieter "noisy" state occured during locked times, so its effect on DARM was examined. As shown in Table 3, there was a slight average increase in DARM during the cluster. This difference can also be seen in the representative spectrum (see Section 6.2) of the cluster

DARM	MIC	ACC
10-13	10-28	1-4
18-22	28-32 (HVAC)	4-10
22-27	32-50	10-28
27-29	50-70	28-32
29-40	70-100	32-48
40-54	100-200	48-60
54-65		60-80
65-76		80-118
75-115		118-122
115-190		122-200
190-210		
210-290		
290-480		
526-590		
590-650		
650-885		
885-970		
1110-1430		

Table 1: Selected clustering BLRMS bands [Hz].

Hz	10-28	28-32	32-50
LVEA	109%	95%	176%
XVEA	87%		89%
YVEA	131%	83%	165%

Hz	32-50	50-70
LVEA	1112%	890%
PLUSX	1183%	1034%
PLUSY	1100%	

Table 2: Percent increase tables (see Section 6.2) for clustered acoustic long, quiet (left) and short, loud (right) states.

Hz	22-27	27-29	29-40
GDS-CALIB_STRAIN	1%	3%	1%

Table 3: Percent increase table (see Section 6.2) for the DARM component of the longer, quieter acoustic cluster.

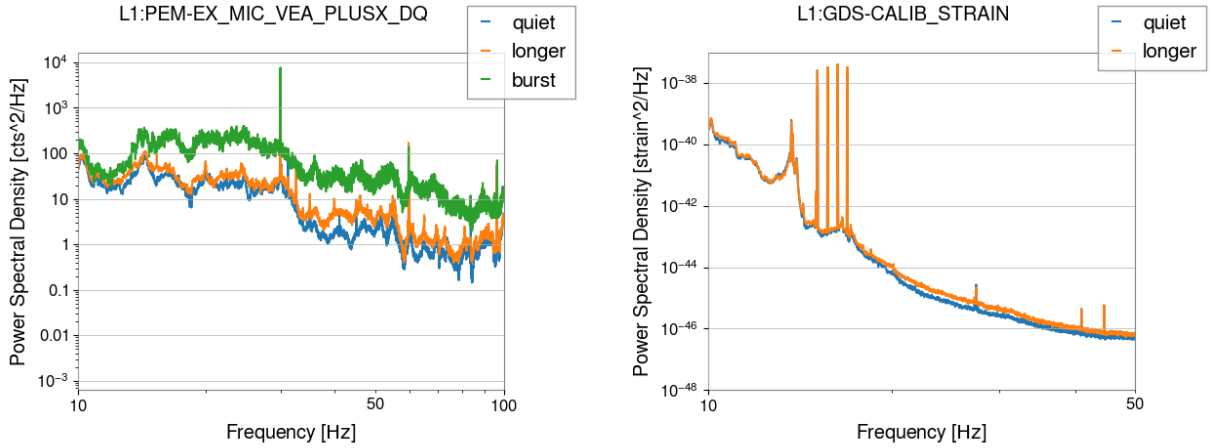

Figure 6: Representative spectra (see Section 6.2) for acoustic states mentioned in Section 5.1. "Quiet" refers to the background "everything else" cluster. The orange cluster seems to affect DARM very little.

(Figure 6). Nevertheless, the noise state does not impact DARM significantly enough to be useful for sensitivity improvements.

5.2 Accelerometers

The "2-hour history" k-means method was used to cluster 30 days of data from various accelerometers from all VEAs in BLRMS bands shown in Table 1. This was done in two groups, one with mostly BSC accelerometers and one focusing on beamtube motion. The primary feature that clustered out in both runs was a 60 Hz motion at the mid stations increasing over half of a day. After making representative spectra (Figure 7), it was concluded that this noise was due to the HVAC system turning on during the warmer period of the day. Motor load causes electrical signals to downconvert slightly as they couple into mechanical movement, so a peak at 58 Hz is characteristic of a fan. The fan movement did not couple into DARM.

Hz	48-60	60-80	80-118
MY VEA BTUBE	256%	259%	123%
EY BSC5 Z	108%		

Hz	48-60	54-65
MY 2100Y BTUBE	618%	
GDS-CALIB_STRAIN		38%

Table 4: Percent increase tables (see Section 6.2) for the 58 Hz accelerometer cluster. There appears to be a correlation to DARM, but looking at the cluster spectrum (Figure 7) reveals this to be due to random fluctuations in the 60 Hz line.

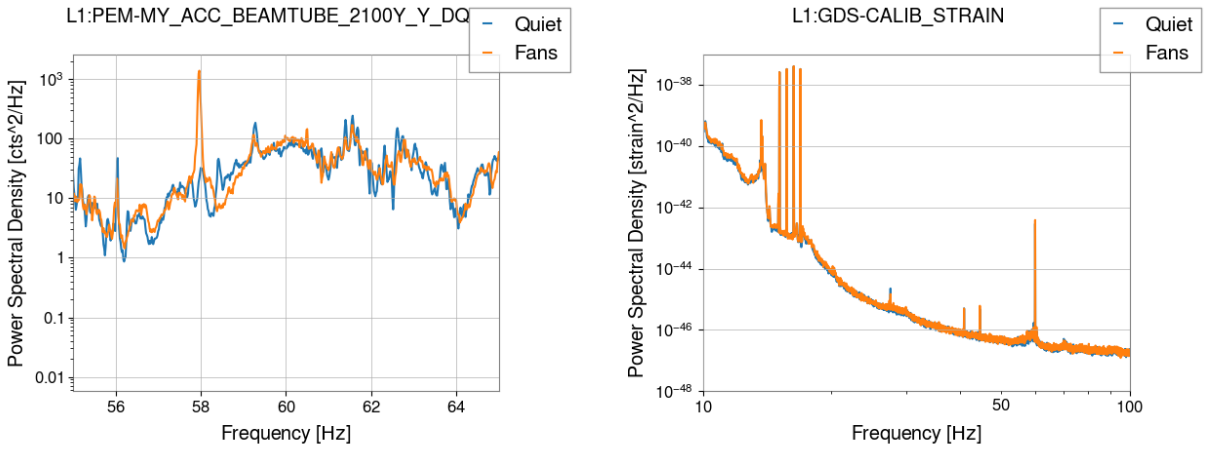

Figure 7: Representative spectra (see Section 6.2) for the accelerometer noise state due to the fans at the mid-stations turning on. This manifests itself as a peak at 58 Hz. There is no coupling to DARM.

6 Developed Code Tools

Many times, scientists are inclined to quickly do their simple calculations imperatively. But time and RAM are not infinite, and with a high-output experiment like LIGO the usual kinds of scripts and operations do not suffice.

6.1 Streaming

One data-wrangling strategy is by programming with a "streaming" rather than "batch" mentality. Many of the gravitational wave search pipelines have the ability to keep up with new LIGO data as it is collected, doing their batch-style work in short chunks, or strides, as the data comes along. This keeps the execution footprint of the program managable, while allowing it to operate on an unending amount of data.

The Python package `GWpy`, used in this project, is the modern equivalent to LIGO's Algorithms Library, a set of common routines designed for LIGO data. Strangely, appending to HDF5 savefiles, a streaming requisite, is not possible in `GWpy` without some lower-level calculations using the `libhdf5` wrapper directly and likely unintented `GWpy` keyword-argument usage.

Two helper functions, `write_to_disk` and `data_exists`, were defined in `util.py` for the purpose of apppending a time series to an existing file and quickly checking the length of saved data without reading it.

A Python module was written to implement the "streaming" idea for any given batch operation. One such task is the computation of BLRMS for channels that don't provide it in DMT frames. The BLRMS-generating function is an implementation of a general `PostProcessor` interface, a `python3.7` dataclass which is fed INI options upon construction (see Figure 8).

Any data processing function which maps an input channel to an output channel and has

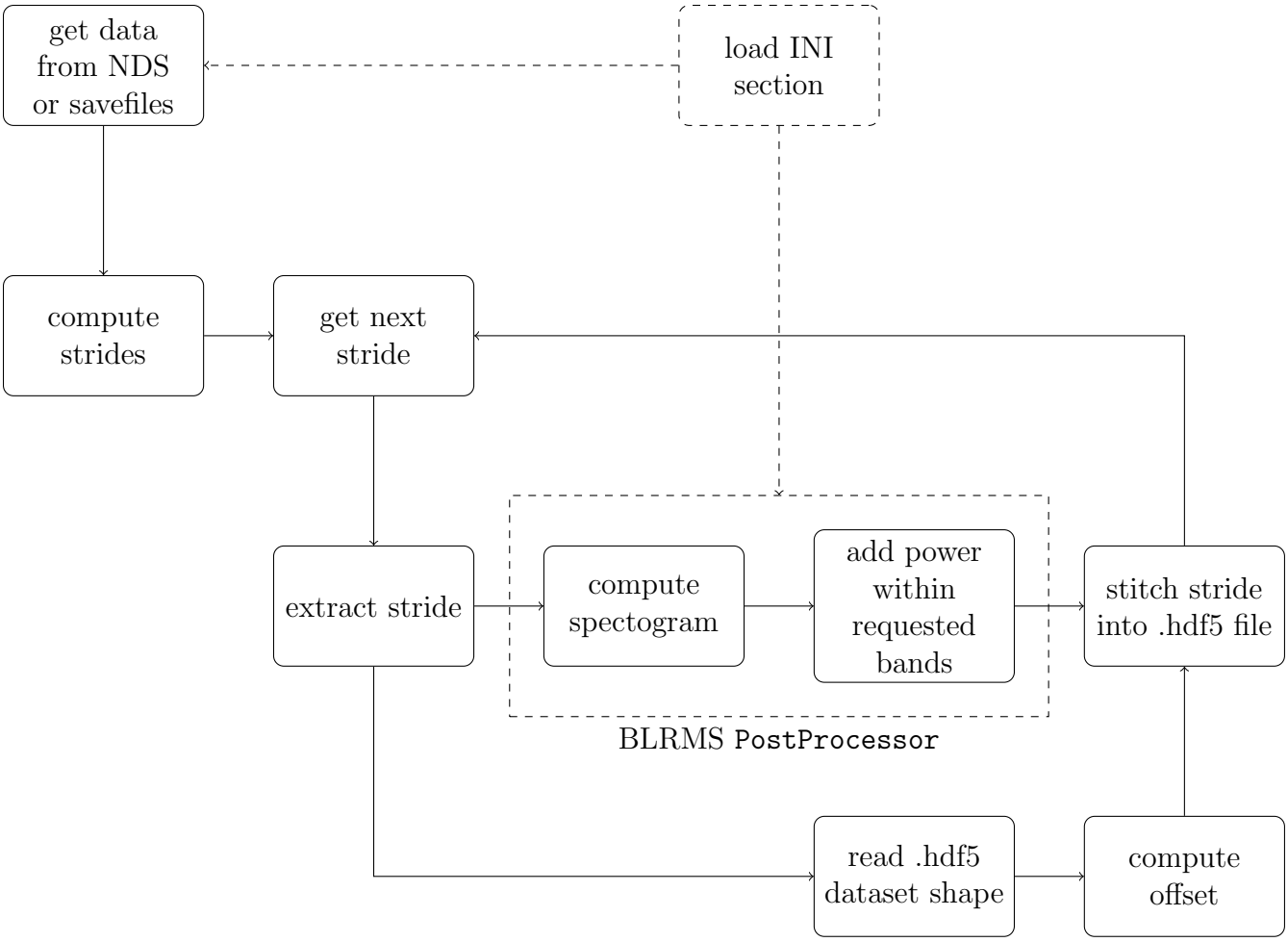

Figure 8: States of the "streaming post-processor" script.

tons of configuration parameters can take advantage of the module by implementing the `PostProcessor` interface. Indeed, converting saved channels to minute-trend and caching NDS downloads in the same format were easy last-minute tasks with this generic structure in place.

6.2 Evaluation of Clusters

The clustering script itself was fairly straightforward, using the high abstraction provided by `scikit-learn`. However, some thought was put into extracting meaning from the clusters.

A script was created to make power spectra for the clustered data representative of each cluster. This is done by taking the median of the 1-minute power spectra for minutes clustered together. An example of this is shown in Figure 10.

Taking the spectra of the clusters allows new states to be categorized without re-clustering, and a way to easily identify frequency conversion that is happening during coupling.

In addition, the script can produce a table that shows which frequency bands are significant in each channel during clustered times. This is done by taking the median percent increase

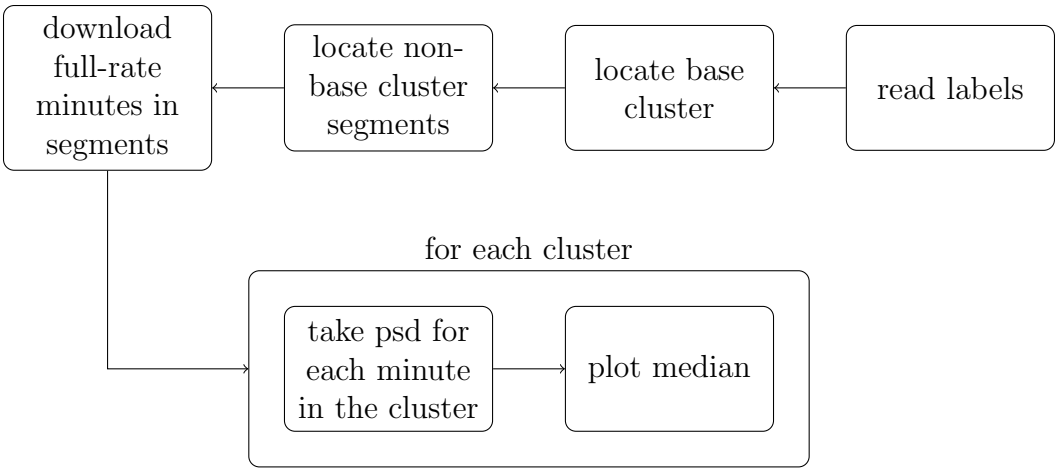

Figure 9: Roughly the states of the "representative spectra" script. The most complicated overlooked detail in this figure is the caching of downloads. The stream writing functions used in the BLRMS-generation script have been moved and are now included from a more general location.

in value of the minute-trend BLRMS of the each channel in each band. The values in Tables 2, 3, and 4 were all produced by this script, but handpicked from the rest of the output for clarity.

Appendices

A Resampling

Downloading and saving full-rate data takes an exorbitant amount of disk space, especially when only a portion of the frequency content is going to be used. This calls for a decimation procedure to be applied to raw downloads before they are saved.

At CIT, Rana mentioned that the default low-pass filtering options in scipy's resampling function produce significant aliasing noise ($> 1\%$) when downsampling by a large factor. According to a test[1] done by Eric Quintero, this issue can be remedied without sacrificing runtime by using (1) a number of FIR taps proportional to the downsampling factor, rather than the default fixed value, and (2) a non-default window (`blackmanharris`).

For a full-rate time series `raw: gwpy.timeseries.TimeSeries`, the fastest and best procedure for resampling to `rate: int [Hz]` would be something like

```
raw.resample(n=20*raw.sample_rate.value/rate, window='blackmanharris', rate=rate)
```

[1] see https://git.ligo.org/NoiseCancellation/GWcleaning/issues/2

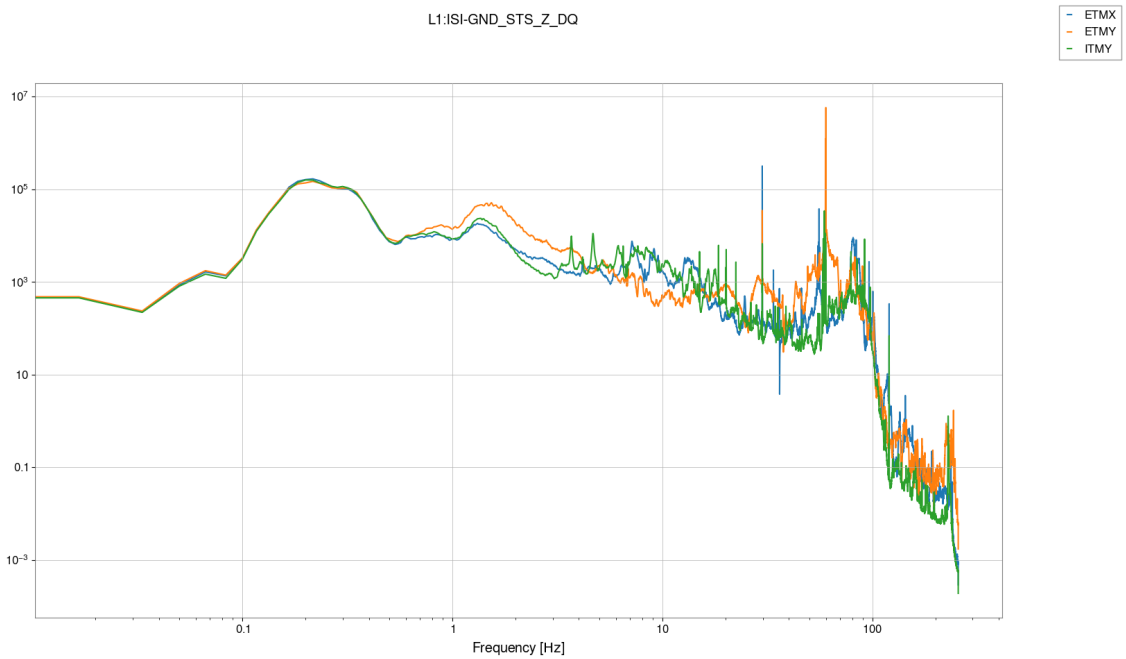

Figure 10: The representative spectrum of a cluster corresponding to train-dominated times. Notice that between 1 and 10 Hz, the seismic motion at ETMY (orange) is greater than at the other VEAs by a factor of about 10.

B Evaluation of Non-k-means Clustering Algorithms

A test[2] which swaps out the k-means algorithm for others implemented in `sklearn` was executed to probe the geometry of the clusters. From a first glance, the Spectral Clustering and Gaussian Mixture algorithms seemed to generalize better than k-means over different feature timescales. However, algorithm upgrades are helpful only after tools for full cluster analysis are in place, and due to time limitations they were not revisited.

References

[1] A. Effler, R. M. S. Schofield, V. V. Frolov, G. González, K. Kawabe, J. R. Smith, J. Birch, and R. McCarthy, Classical and Quantum Gravity **32**, 035017 (2015).

[2] LIGO Document T1700198-v1

[3] aLIGO LLO Logbook entry 45374 by Gabriele Vajente

[2] https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html