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

Technical Note **LIGO-T1900287–v1** 2019/07/15

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 amplitude scales of astrophysical GWs, 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 treatment 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. This is used for a multitude of purposes, including the data quality report (DQR) used for time segment vetoing, based on direct coherence of PEM channels to DARM. Supplementing coincidence analysis between the two detectors, DQR prevents GW-like noise transients from being falsely categorized as events. While vetoing and determining signal quality is useful, detector livetime 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 generally 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[1]. 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

- As a primary goal, **algorithms or clustering approaches which correctly identify known noise events need to be found**. As every algorithm has inbuilt assumptions about the dataset it is applied to, the results of an algorithm performance test on labeled data will yield information about the structure of the data. The general temporal non-stationarity of the DARM noise will need to be accounted for by varying testing time windows.

- The secondary goal is to **create a clustering approach to discover previously unknown noise correlations and possibly sources**. This is where the “detector characterization tool” that this project aims to advance will be functional—revealing new noise coupling pathways will help identify ways to improve the detector sensitivity.

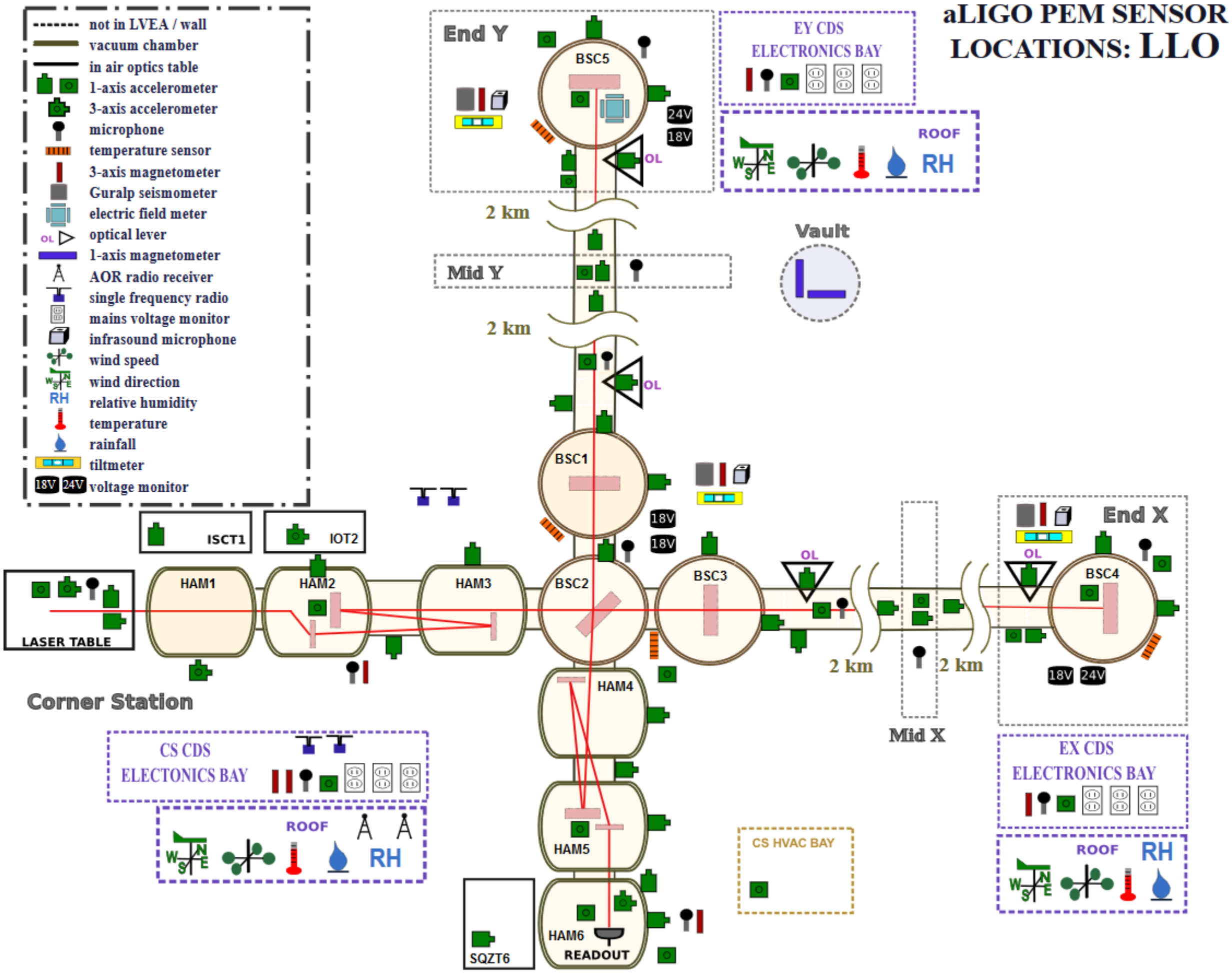

Figure 1: Schematic PEM map at the LIGO Livingston Observatory (L1). Shaded areas are in vacuum.

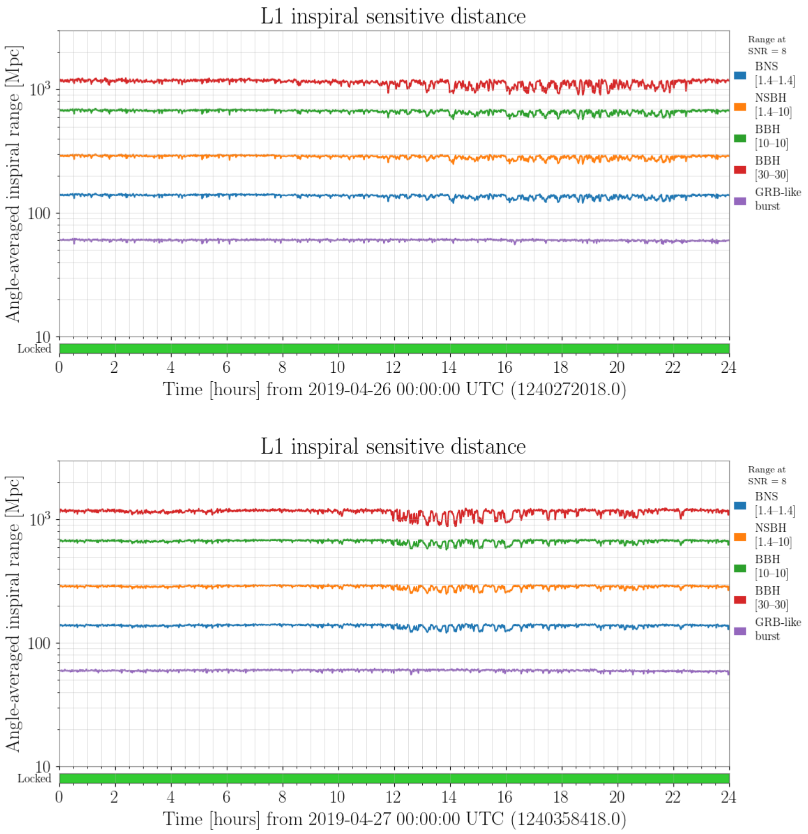

Figure 2: Detector range at L1 seems to consistently reduce during the day (∼6am-5pm CST). For large BBH in these plots, the reduction is about 300 Mpc. The source of this has been pinpointed to the Y end station, but the mechanism isn't fully clear.

3 Approach

Initially, a program will be written to take the spectral power of any PEM channel, in the form of band-limited RMS (BLRMS), likely using established methods like looping through a smoothed spectogram of the channel[2].

To reach the first objective, a modular `python` testing suite will be written to probe the structure of the multidimensional frequency-domain sensor data. This will strategically implement `scikit-learn` clustering algorithms and classifiers with different optimal regimes of function or working assumptions and evaluate them using point labeling. This will require, additionally to researching clustering or unsupervised classification algorithms, thinking of as many variables which may affect the data structure (such as looking at different time windows) and intelligently testing them. Optimizations will need to be considered so that run times are reasonable.

The program tackling the second objective will use working clustering approaches identified in the first objective to find new noise correlations. In the event that no individual algorithm or technique outperforms the rest for all types of sensory data, the final program will use the modular programming environment created for the testing suite to match techniques to the regimes that they work in. The structure of the input data as determined by the first objective, including the dimensionality probed by the extra variables, may lend itself to additional algorithms that can be used to combine the target regimes. To this end, extra algorithm research will be conducted with specific consideration of the solved structure.

4 Interim Report 1

In the first three weeks of the project, clustering scripts and BLRMS-generating scripts were developed and tested. Testing the clustering code on a dataset with known sources of noise was thought to be a good first-order check of its efficacy.

4.1 k-means clustering with histories

The k-means algorithm was used to cluster the two hours of minute-trend seismic BLRMS preceding each point in time. Thus, each coordinate in the clustering sub-space for a channel was as follows:

$$\{s(t_0), s(t_{-1}), s(t_{-2}) \cdots, s(t_{-n})\} \tag{1}$$

with $s(t)$ the seismometer velocity at time t. Each dimension corresponded to “channel value a specific number of minutes ago”. This allowed trends over time to be matched together in a phase-agnostic way.

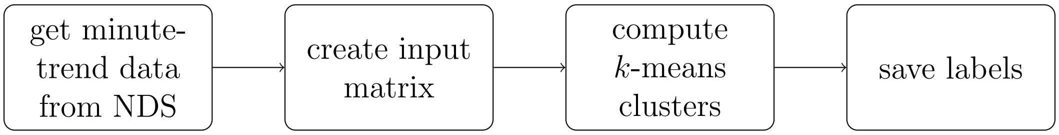

Figure 3: Clustering script flowchart.

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 (see Figures 4-7):

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

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.

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 re-run with only anthropogenic seismic BLRMS bands at the end stations, which yielded very clear distinction between the anthropogenic noise types (Figure 8).

A test[1] which swaps out the k-means algorithm for others implemented in `sklearn` is being executed to probe the geometry of the clusters. At the time of submission, the job has not completed.

4.2 BLRMS-generating script

A script to generate minute-trend BLRMS from raw PEM channels was created. This will enable BLRMS clustering for those channels not already in BLRMS frames.

Some thought was put into this script. It is designed to post-process a channel in real time, using strides and HDF5 appending to provide a "streaming mode". The BLRMS-generating function is an implementation of a general post-processing interface, a `python3.7` dataclass which is fed INI options upon construction (see Figure 9). Any data processing function which maps an input channel to an output channel and has tons of configuration parameters can take advantage of the streaming mode developed in this script by implementing the `PostProcessor` interface.

Considerable effort was expended to keep both the BLRMS and clustering scripts `GWpy`-compliant. Enabling HDF5 appending in `GWpy` savefiles and modifying `GWpy` plots seem like common tasks for users of `GWpy`, but they needed some hackery and keyword-argument abuse (i.e. making use of possibly unintended library behavior) to work properly.

4.3 Next Steps

The next step is to cluster with accelerometer, microphone, and DARM BLRMS. There are a number of strategies that will be employed in light of the past few weeks' results.

For instance, the clustering algorithm appears to gravitate toward differences across a channel parameter that is more varied in the input. The initial run with 6 BLRMS bands and 3 sensors distinguished between noise in different bands more than noise in the same band but in different sensors (i.e. trains vs. day/night noise). When the number of sensors was similar to the number of bands, the sensor-specific noise was more clearly parsed out.

For unknown noise, it will thus be important to run multiple tests on the same data which focus on each varying parameter. That is, it may be best to cluster all sensors in one band, and all bands of one sensor, separately, rather than doing it all together.

In addition, much will rest on appropriate BLRMS band selection. Instead of simply using half-order-of-magnitude spacing, it may be worthwhile to make a program that searches for time variance in the channel by frequency, so as to pick the most varying, and possibly charactaristic, bands for the channel. In his BLRMS implementation, Vajente includes hard-coded DARM bands, so it may be smart to stick with those for DARM[2].

After experiencing the most naive presentation of the clusters, namely, time series plots with colored points, it occurs to us that there should be a way to programmatically identify the pattern that characterizes each cluster, to enable looking into mechanisms and solutions for the noise. This will probably involve getting the frequency of occurrence of the cluster, as

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

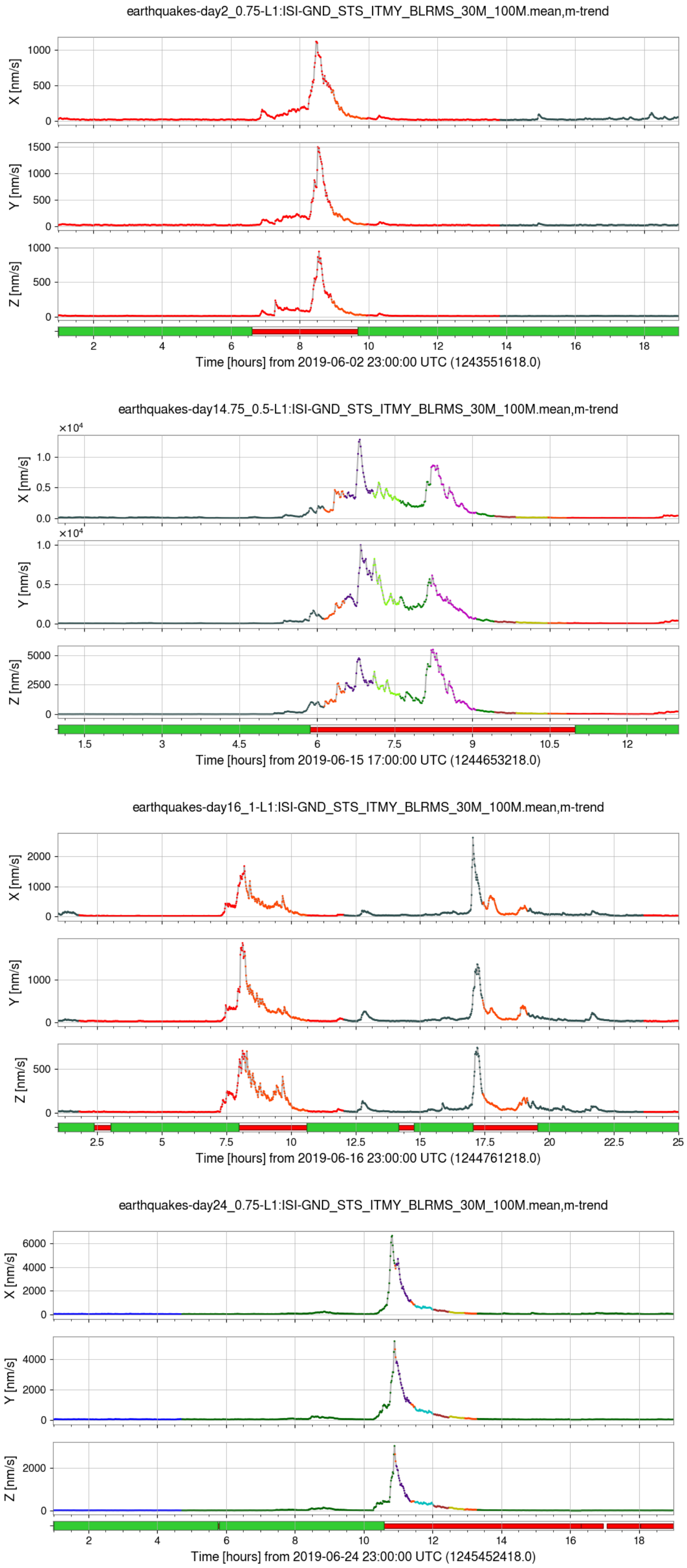

Figure 4: Examples of earthquakes. These vary most of all seismic noise events, so most of the extra, unrequired clusters tend to pick details out in these. For instance, two different kinds of earthquakes were clustered (here orange and purple predominantly).

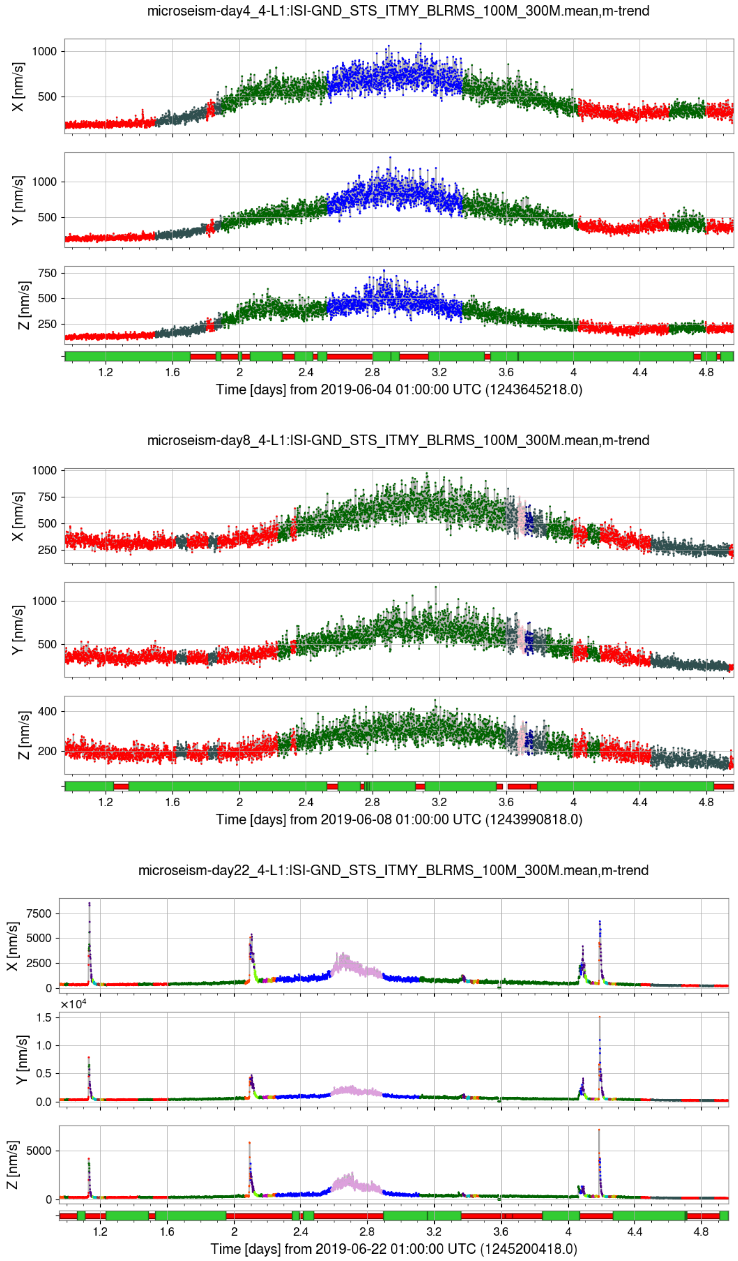

Figure 5: Examples of microseisms identified by the k-means algorithm. Because the algorithm was run with more clusters than needed, several clusters are assigned to this class of event. Running with less clusters fixes this issue, but details are sometimes missed.

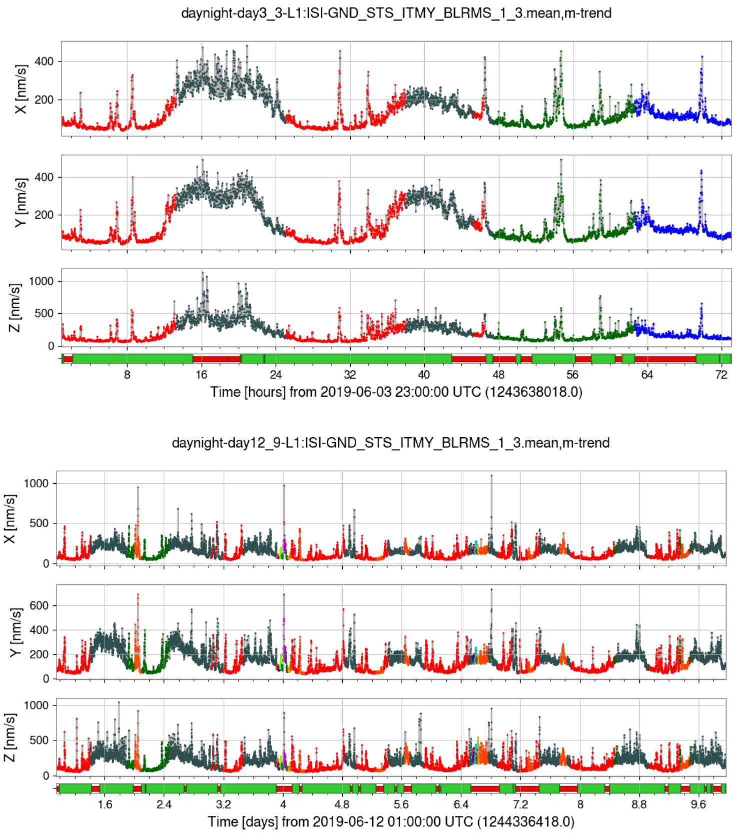

Figure 6: Day-night anthropogenic noise variation is identified by the *k*-means algorithm. These clusters are depicted gray.

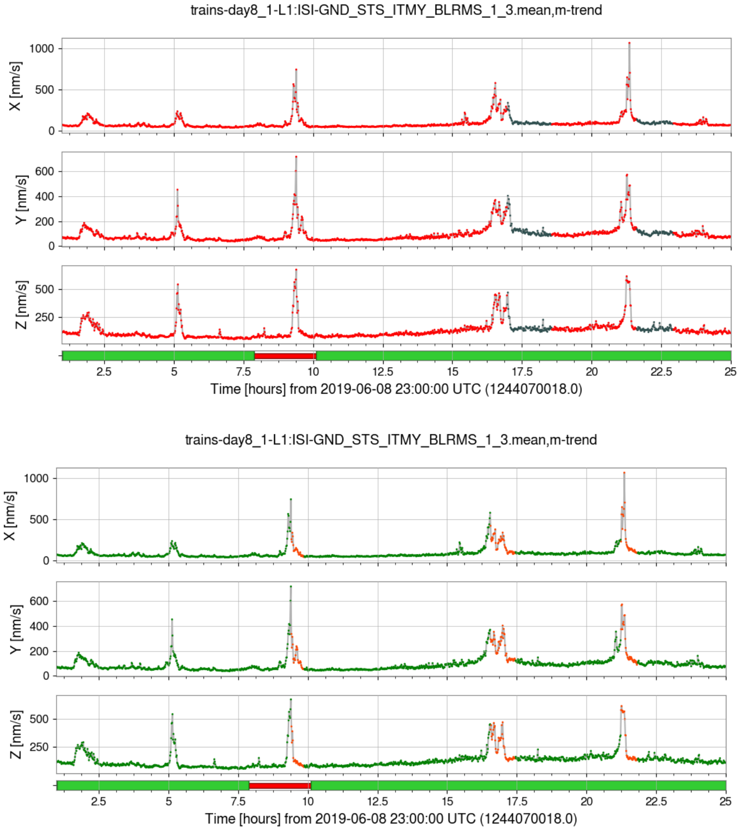

Figure 7: Trains are clustered with the day/night anthropogenic noise. Shortening the history window from 2 hours (upper) to 30 minutes (lower) helps to clarify this.

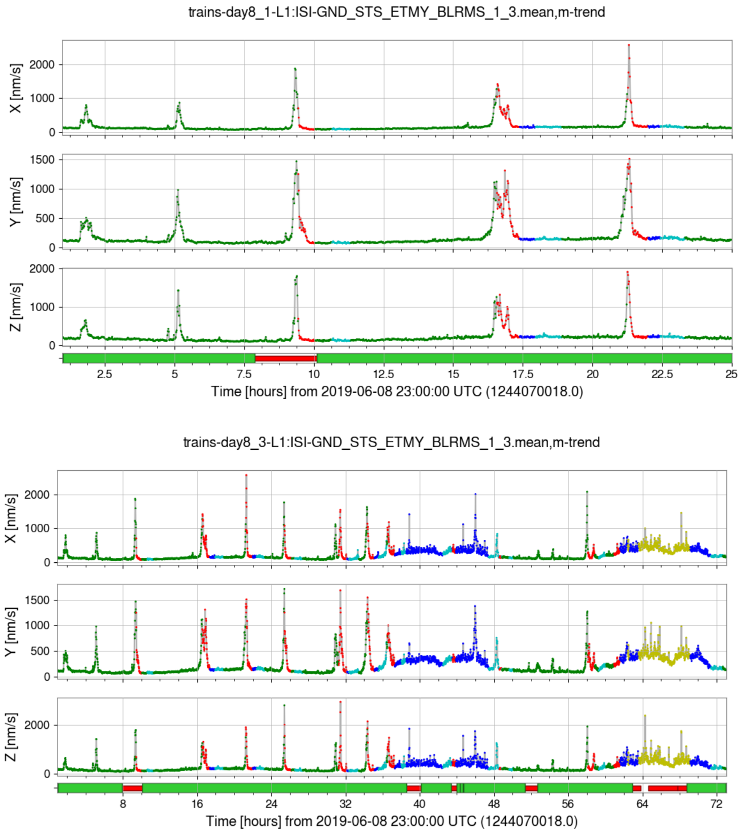

Figure 8: Here, the clustering space was reduced to only the anthropogenic noise bands. This enabled identification of trains (red) separate from day/night noise (blue).

well as its average dimensions (in time and amplitude), and ratios of presence in bands and sensors.

References

[1] LIGO Document T1700198-v1

[2] aLIGO LLO Logbook entry 45374 by Gabriele Vajente

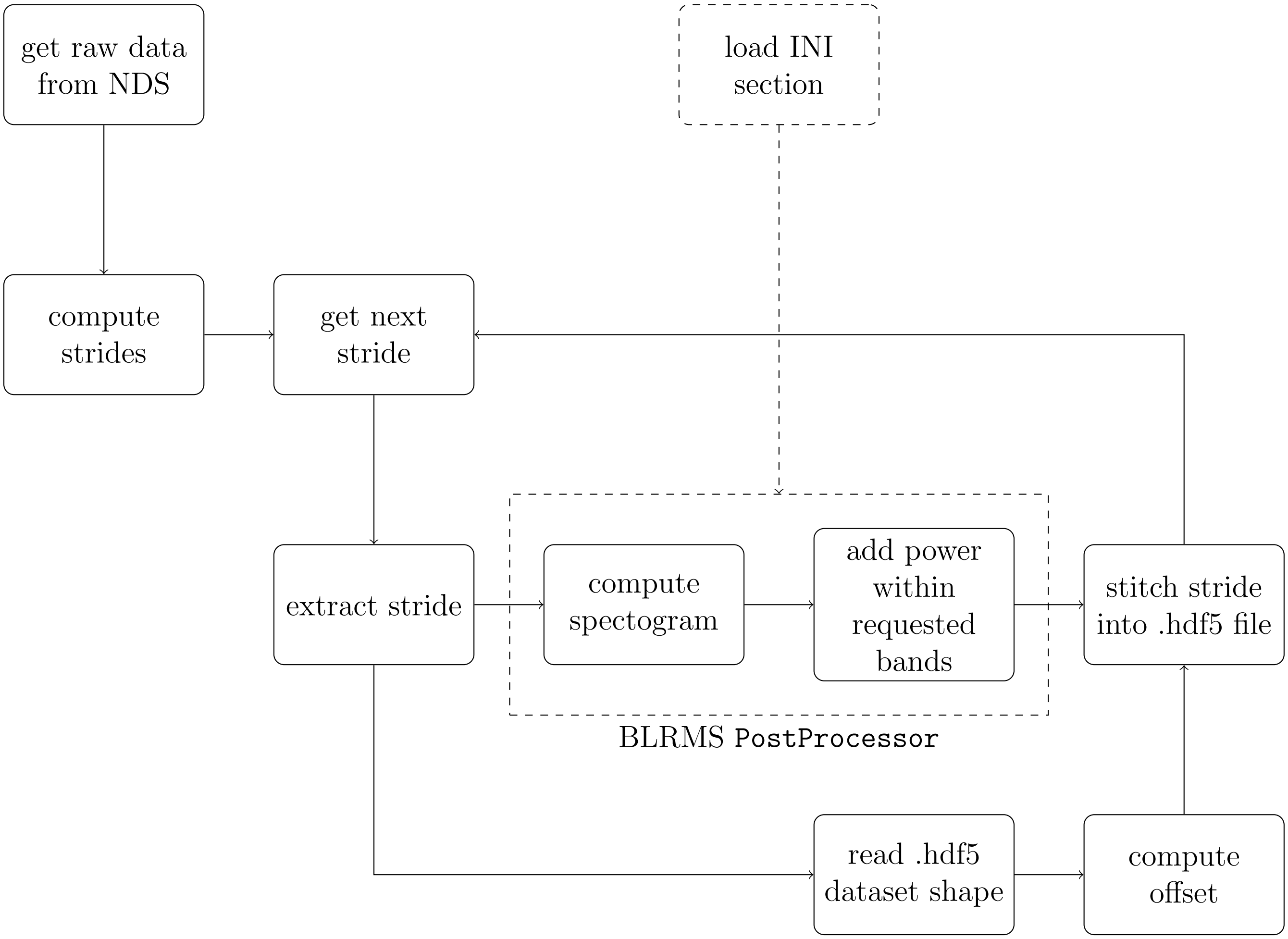

Figure 9: States of the “streaming post-processor” script.