Implementing New Veto Analysis Methods in the PyCBC Search for Compact Binary Coalescences

LIGO Caltech SURF Program 2020, Mentor: Derek Davis

Brina Martinez[1]

[1]University of Texas Rio Grande Valley, Brownsville, TX 78520, USA

E-mail: `brina.martinez@ligo.org`

Abstract. The PyCBC pipeline has been used since the first detection made by LIGO and continues to be used today. Some issues arise with current methods of veto analysis in the PyCBC search for a gravitational wave including the possible removal of a hidden signal, an increase in how long a search takes, or a decrease in analyzing how significant a detection is. We describe the steps we will take in tackling the current status of PyCBC and how we plan to improve veto analysis methods in the search from compact binary coalescences.

1. Introduction/Background

The Laser Interferometer Gravitational-Wave Observatory (LIGO) [1] led to the discovery of gravitational waves (GW) with their first detection of a binary black hole (BBH) collision GW150914 [2]. Since then, there have been three separate observing runs which have included detector improvements resulting in an increase in sensitivity leading to multiple significant events being discovered on a weekly basis in O3. So far, 13 confident detections have been announced [3][4][5].

The PyCBC pipeline is one of the many pipelines that has been used since the first detection made by LIGO [6]. PyCBC is used to find GW events that are produced by a compact binary coalescence (CBC) and determines how significant each event is. PyCBC uses match filtering to compare and match triggers (signals) against templates (or models) that contain multiple different waveforms of which a GW detection should look like and re-weighs the relationship with the estimated power spectral density (PSD) of the detectors involved. PyCBC also uses gating, coincidence tests, and measures the false alarm rate (FAR) of PyCBC pipeline measured events [7]. The time-slides portion of PyCBC's current methods assesses how significant our event is. A detection seen by both LIGO detectors can help us calculate a network signal-to-noise ratio (SNR) in which we can determine our FAR. The FAR helps us determine the likelihood of seeing our detection again within the same network SNR, so the lower the FAR the less chance our detection was due to noise. Our SNR used in match filtering is helpful in extracting our signal from gaussian noise.

With the amount of triggers, both strong and weak, that are produced before and after match filtering we can sometimes find loud glitches that are short in duration sneak across data quality tests. This leads to an increase in analysis time and results in a decrease in search sensitivity through two mechanisms, dead-time and ringing of the match filter [7][8]. As detections become more frequent, the quality and confidence of these detections need to increase. This project, if successful, will be automated and implemented in the future observation runs by LIGO as a data quality tool to improve the search for GWs.

2. Objectives

In this project we plan to improve the volume of searches in both strong and weak signals by implementing new veto techniques and new data quality flags. One method PyCBC currently uses is data quality flags and time-slides. This method looks at a segment of data and vetoes out noise using data quality flags determined by noise in the auxiliary data segment. The problem with this is current data quality flag methods are not as good as we want them to be, as they can remove a potential detection of a GW signal if it is covered by glitches.

With the improvement of PyCBC veto techniques we will be able to better differentiate signals from noise in the detector and improve how long searches take. The two main goals we want to assess within our improved data quality flags are:

- We want to remove as many glitches as possible to increase the significance of signals.
- We want to remove as little time (data) around a segment to reduce the chance of accidentally removing a GW signal.

Though these two main goals are a bit incompatible with each other since one wants to keep more time segments and the other wants to remove as many glitches possible, we have to take a look at our volume-time (VT) which is represented by:

$$\mathrm{VT} = [\text{sensitive distance}]^3 \times [\text{analysis time}] \tag{1}$$

If we achieve even the slightest increase in sensitive distance we can correlate with an increase in search volume. To achieve these results we will need to measure and calculate how the PyCBC search responds to different configurations, vetoes, and ranking statistics. We will look at how the VT and recovered ranking statistic are affected before and after our chosen methods. Once we see an improvement in searches, we will know which direction we should continue to follow. The tools we determine to work best will be implemented into the PyCBC search pipeline and will work continue to increase the VT of the search to find more GWs in future observing runs.

3. Approach

To accomplish our objectives I will begin by familiarizing myself with how PyCBC works, how I can run our tests, and how to interpret our results. This will probably take a bit of the first two weeks or weeks before the official start date. We will be using multiple tools including Gravity Spy [9], Hveto [10], and iDQ [11] to develop data quality flags and vetoes. Another step we will take is to test our own new methods, see how signals react and are recovered, both weak and loud, and how our VT is affected.

3.1. Approach Example

In the example we can see an example of PyCBC using the time-slides method with data quality flags. The plots below show us the FAR V. SNR of data containing glitches and a known signal. In Figure 1 we can see our FAR V. SNR plot of the data before applying any data quality vetoes. Our data without data quality vetoes gave us a FAR of 0.4101 per hour.

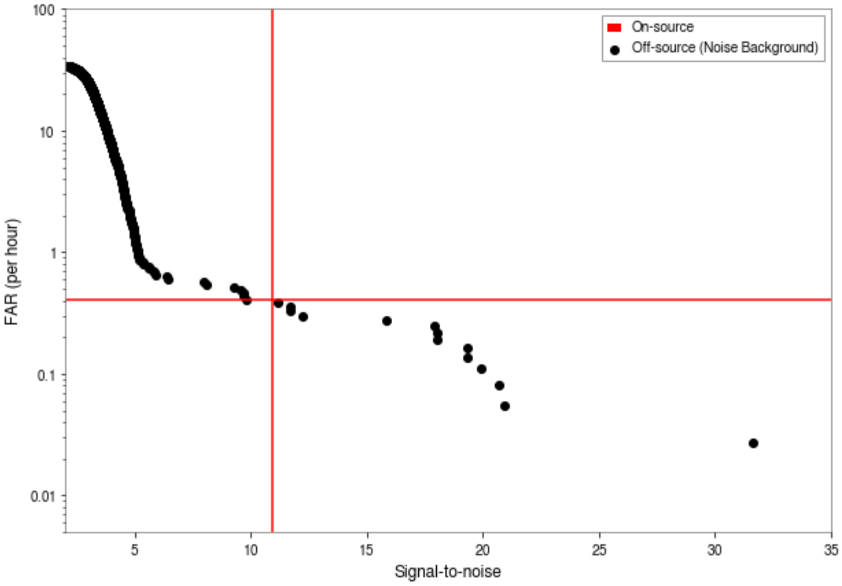

Figure 1: Data containing glitches

In figure 2 we applied a data quality flag to the same data as before except we whitened the auxiliary data and set a threshold and produced windows around glitch times identified. This led to our signals becoming more significant with a FAR of 0.0567 per hour, a ratio of distance of 3.97, a ratio of time of 0.69, and a ratio of VT of 43.54. We saw an improvement in our data where our signal is much louder than our glitches but we can keep testing new methods and parameters to try to find even better results.

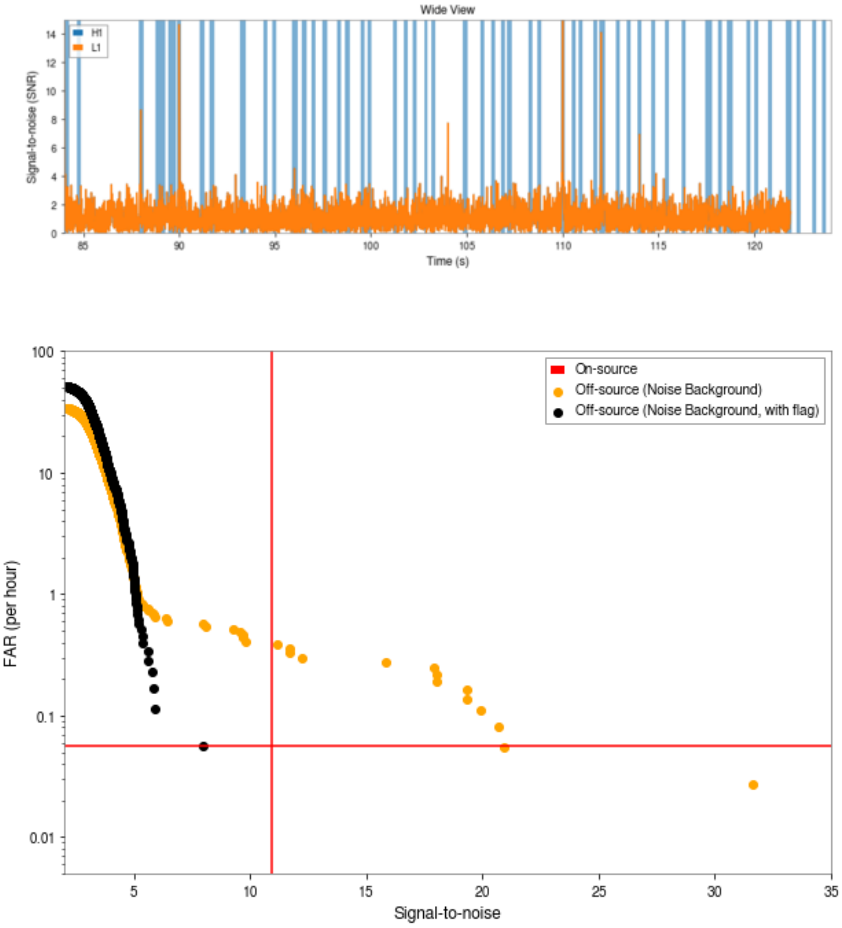

Figure 2: Data with data quality flag

In figure 3 we applied a band-limited root-mean-square (BLRMS) to the data that focuses on the main power frequencies contained in the glitches, along with new data quality flag threshold and window values from before to better identify the glitches. This led to our signals becoming more significant with a FAR of 0.0319 per hour, a ratio of distance of 5.40, a ratio of time of 0.93, and a ratio of VT of 145.78. This result in having a small amount of time lost and an increase in sensitive distance led to an enormous jump in our VT ratio. Applying the BLRMS and new data quality flags we were able to improve our SNR for our signal and glitches even further.

We will mainly be working between each other but if necessary we will be collaborating with the Detector Characterization and PyCBC groups in LIGO. Completion of this project and any results we obtain will mostly be dependent on our performed tasks and chosen methods in which we will be applying to accessed data from previous observation runs.

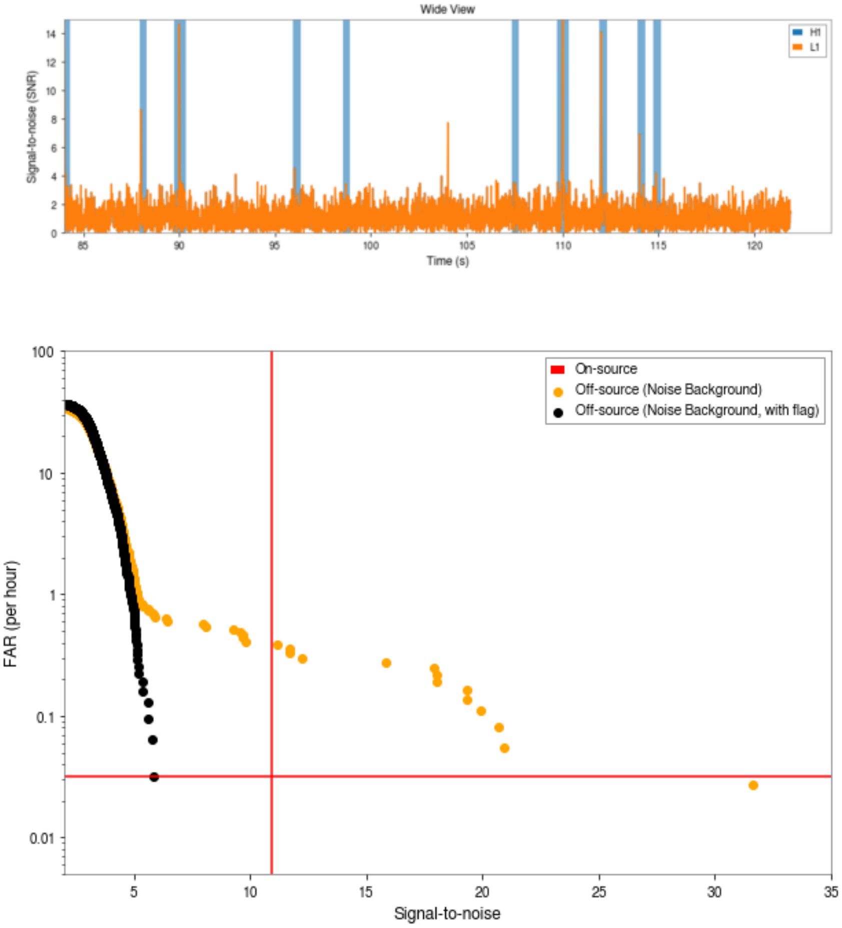

Figure 3: Data whitened with BLRMS and new data quality flag

4. Work Plan/Schedule

Before official start date:

- Watch videos and read material to familiarize myself with the background of PyCBC, Hveto, data quality, and anything pertaining to our main goal.
- Practice implementing possible techniques to become familiar with the environments.

Weeks 1-2:

- Become familiar with using PyCBC and learn to recognize first hand what needs to be improved.

Weeks 3-4:

- We will begin to apply different techniques that we believe can improve our search volume and response such as applying different vetoes, using Hveto, and different data quality flags.

- Produce plots needed to analyze results.

Weeks 5-6:

- Continue running tests from different techniques and see what does and does not work.
- Review how ranking statistics, sensitivity, volume, and search has changed.
- Develop vetoes, both source-specific and non-binary.

Weeks 7-8:

- We will continue tests and review results from previous weeks up until now and decide whether or not techniques were effective and continue to apply methods and run tests.

Weeks 9-10:

- Finalize results and gather all information which includes plots, data.
- Produce a final report that includes the nature of our project and its objectives, the methods we implemented, any figures that are related to our testing and results followed by references and acknowledgements.
- Produce a final presentation that will be 15 minutes and includes information on the project and why it is important, methods we implemented and how we did them, and finally the results obtained and ideas for future work. Acknowledgements included at the end.

5. Acknowledgments

Computing support for this project was provided by the LDAS computing cluster at the California Institute of Technology. LIGO was constructed by the California Institute of Technology and Massachusetts Institute of Technology with funding from the National Science Foundation, and operates under cooperative agreement PHY-0757058. This work carries LIGO Document number T2000349-v1.

References

[1] J. Aasi et al. Advanced LIGO. *Class. Quant. Grav.*, 32:074001, 2015.

[2] B.P. Abbott et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.*, 116(6):061102, 2016.

[3] B.P. Abbott et al. GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs. *Phys. Rev. X*, 9(3):031040, 2019.

[4] B.P. Abbott et al. GW190425: Observation of a Compact Binary Coalescence with Total Mass $\sim 3.4 M_\odot$. *Astrophys. J. Lett.*, 892:L3, 2020.

[5] R. Abbott et al. GW190412: Observation of a Binary-Black-Hole Coalescence with Asymmetric Masses. 4 2020. arXiv:2004.08342.

[6] B.P. Abbott et al. GW150914: First results from the search for binary black hole coalescence with Advanced LIGO. *Phys. Rev. D*, 93(12):122003, 2016.

[7] Samantha A. Usman et al. The PyCBC search for gravitational waves from compact binary coalescence. *Class. Quant. Grav.*, 33(21):215004, 2016.

[8] B P Abbott et al. Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO's first observing run. *Class. Quant. Grav.*, 35(6):065010, 2018.

[9] Michael Zevin et al. Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science. *Class. Quant. Grav.*, 34(6):064003, 2017.

[10] Joshua R. Smith, Thomas Abbott, Eiichi Hirose, Nicolas Leroy, Duncan Macleod, Jessica McIver, Peter Saulson, and Peter Shawhan. A Hierarchical method for vetoing noise transients in gravitational-wave detectors. *Class. Quant. Grav.*, 28:235005, 2011.

[11] Reed Essick, Patrick Godwin, Chad Hanna, Lindy Blackburn, and Erik Katsavounidis. iDQ: Statistical Inference of Non-Gaussian Noise with Auxiliary Degrees of Freedom in Gravitational-Wave Detectors. 5 2020. arXiv:2005.12761.