Effects of Different Data Quality Veto 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 search pipeline has been used since the first gravitational wave detection made by Advanced LIGO and continues to be used today in the search for gravitational waves. To identify gravitational waves from compact binary coalescences, PyCBC runs a matched filtering and chi-squared (χ^2) consistency test to determine significant signal-to-noise ratios and compares triggers to previously modeled templates. To confidently detect gravitational waves, we need to mitigate noisy data, which in return improves the sensitivity of searches. Current veto methods use data quality flags to veto and remove triggers in LIGO data that are believed to have terrestrial origin, though these methods risk accidentally removing signals and must be finely tuned to prevent a decrease in the search sensitivity. In this investigation, we test different veto methods based on the current set of data quality flags and detector characterization tools. We analyze how simulated signals are recovered by the PyCBC pipeline and the overall change in the sensitivity of the pipeline. Our results show an improved veto method that increases the significance of signals and the overall number of detectable signals without removing data. The results of this investigation can be implemented in the PyCBC search pipeline in future observation runs held by LIGO as a data quality tool to improve the search for gravitational waves from compact binary coalescences.

1. Introduction and Background

The Laser Interferometer Gravitational-Wave Observatory (LIGO) [1] discovered gravitational waves (GW) with the first detection of a binary black hole (BBH) collision, GW150914 [2]. Since then, there have been three separate observing runs along with detector improvements that have increased the rate of detections to multiple times per week in the third observing run (O3). So far, 14 confident detections have been announced [3][4][5][6].

The PyCBC [7][8][9][10][11][12] pipeline has been used since the first detection made by Advanced LIGO [13]. PyCBC identifies GW events that are produced by compact binary coalescences (CBCs) and determines how significant each event is as compared to the noise in the detectors. PyCBC uses matched filtering to compare the data against templates that model what GW detection should look like and re-weighs the relationship with the estimated power spectral density (PSD) of the detectors involved. Matched filtering calculates the signal-to-noise ratio (SNR) of our templates and PyCBC then uses a χ^2 test to filter our SNR and remove triggers that do not match our templates very well. PyCBC uses gating and coincidence tests to measure the false alarm rate (FAR) of recorded events [11]. PyCBC's time slides method is used to generate simulated background data with the two LIGO detectors. A detection seen by both LIGO detectors can help us calculate a network SNR in which we can determine a FAR. A significant FAR helps us determine the likelihood of seeing our detection again with the same network SNR, so if our FAR is low or decreased the less chance our detection is due to terrestrial noise.

With the amount of triggers that are identified using matched filtering, we sometimes find loud, short duration glitches sneak across data quality tests. This results in a decrease in search and ringing of the match filter [11][14]. As detections become more frequent, the quality and confidence of detections need to increase. In this investigation I am analyzing veto analysis methods that will remove or re-rank as many glitches as possible from the data using data quality (DQ) flags to increase the significance of signals. This investigation analyzes the aspects of removing as little data as possible to reduce the chance of accidentally removing a gravitational wave signal. These methods will be applied to PyCBC triggers and be used to calculate the change in background and sensitivity of the search. We will evaluate how the probability and distribution of triggers change with respect to time and how we can focus on times that are interesting to improve PyCBC. We will also analyze how the PyCBC search responds to different configurations and DQ flags. Once we see an improvement in searches, we will know which direction we should continue to follow. This project, if successful, will be automated and implemented in the PyCBC search pipeline along future observation runs by LIGO as a data quality tool to improve the search for GWs.

2. Work Plan/Schedule

From the beginning of this project until recently, a few changes in the timeline have occurred regarding steps we will take to tackle our goals and when we will take them and that is due to a better understanding of where we stand currently with investigations and how they are flowing.

Before official start date: I watched videos and read papers to familiarize myself with the background of PyCBC, Hveto, data quality, and anything pertaining to our main goal.

Weeks 1-2: I Became familiar with using PyCBC and importing data and learned to recognize first hand how to analyze plots, and figure out what needs to be improved when necessary. I worked on investigations regarding the parameters PyCBC works with such as likelihood, FAR, and SNR significance.

Weeks 3-4: I continued to work on investigations to understand the data I work with and began practicing the use of DQ flags. I produce plots needed to analyze results and work on the first interim report.

Weeks 5-6: I took what I had previously investigated and used that information together to develop vetoes and generate time slides to analyze the change in background.

Weeks 7-8: I am continuing to run tests of how our veto methods affect the background, ratios of distance, ratios of time, and ratios of volume*time (VT). I am producing plots and gathering information for the second interim report.

Weeks 9-10: I will 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 investigations 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 utilized them, and finally the results obtained and ideas for future work. Acknowledgements included at the end.

3. Progress

The first few weeks of the SURF program, I worked on becoming familiar with PyCBC, statistics we would need to use, data quality terms and flags, and other parameters that relate to our goals. This included learning how to understand the data I produced, such as plots, and how to apply small details from different investigations into the overall project.

3.1. PyCBC time-slides and Data Quality flags

Before my official start date in June I familiarized myself with PyCBC by investigating matched filtering, DQ flags, and time slides to remove glitches and reveal hidden signals with simulated data. Time slides work by sliding data from one detector against data of the other, in our case the Hanford (H1) and Livingston (L1) detectors. Time slides is an efficient method to generate background data between the two detectors since we understand glitches occur at different times which shows us they cannot be from a GW signal. DQ flags are segments of time singled out that contain glitches we do not want since they correlate with problematic noise in the detector. Problematic noise makes it difficult to run analysis on signals and decreases their significance. When we identify these glitches, we are able to reduce their impact on the search and make our signal more significant. There are different ways we can set up a DQ flag and choose how we want to single them out, but in this investigation we will be looking at whitened auxiliary data above a threshold and with specified windows of time segments to identify our peaks. In this investigation, I am analyzing a data set that includes simulated random Gaussian noise that is recolored to mimic noise properties we could find in either one of the LIGO detectors. The data includes three different types of injections to analyze:

- Simulated sine-Gaussian bursts, which are similar to common, short duration glitches present in LIGO data.
- Simulated gravitational-wave signals with a limited bandwidth, to represent a 'worst case scenario' glitch.
- Simulated gravitational-wave signal, which we will be trying to identify from our filtering and DQ flags.

In this investigation we want to not only calculate and recover a significant signal but also calculate a significant FAR.

Figure 1 shows whitened auxiliary data which include our signal along with glitches that have very high significance and loud SNR's. From looking at the time series, we are able to pick a threshold and window size we believe will identify peaks and generate a DQ flag. Once we apply our DQ flag to the time series we are able to generate time slides, re-plot our original data, and compare it to the original curve that does not contain our DQ flag. After applying our DQ flag, we can see our signal becomes louder than any of the glitches and background.

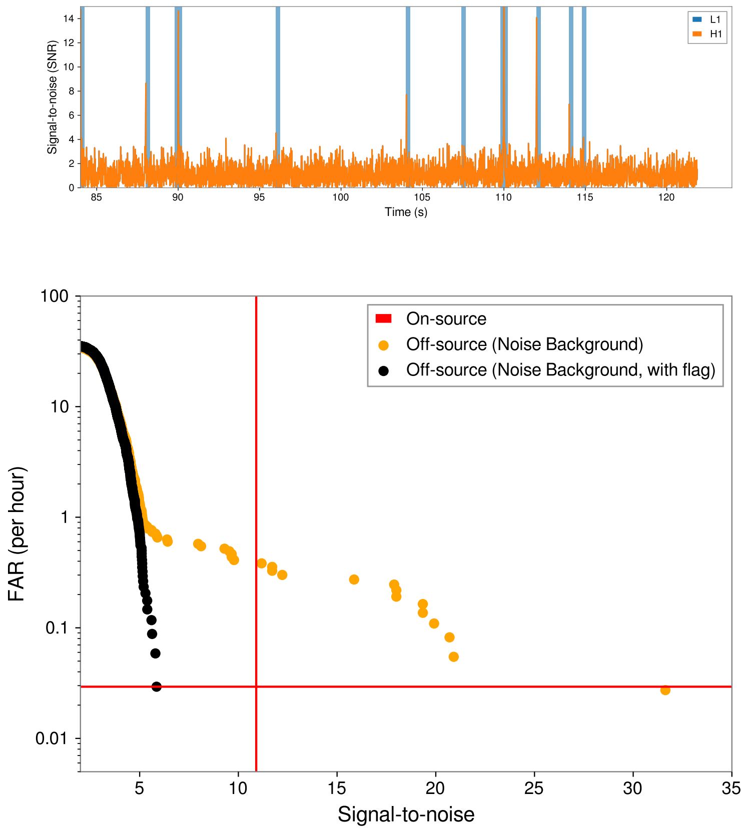

Figure 1: In the plot we are able to see a time series containing data with glitches and a signal which we are not able to tell apart. To help our signal SNR become more significant than the SNRs of our glitches, we applied DQ flags and used time slides to find coincidences between the data from our two detectors H1 and L1. We then established a threshold where the whitened auxiliary data is above the value we choose, and we establish windows to highlight those peaks, making sure not to highlight our signal. Our results compare our data before and after applying the DQ flag. We can see a significant increase in significance for our signal compared to the background. In the time series we can see blue bars highlighting the peaks that fall into our DQ flag which we want to remove without highlighting our signal which is also a significant peak. In the SNR vs FAR plot we can see our original background in orange that was extremely loud compared to our signal and our new background in black that significantly reduced and made our signal the loudest part of our data. Our original FAR was 0.3526 per hour. After applying our time-slides and DQ flags our FAR is now 0.0219 per hour, significantly lower.

3.2. Likelihood

In my first week of SURF I ran an investigation to familiarize myself with how likelihood functions work and how they are used in PyCBC. Likelihood shows us the probability of how often an outcome, which we can label x, is expected to occur in a given model, which we can label θ.

$$\mathcal{L}(\theta \mid x) = p_\theta(x) \tag{1}$$

This outcome can occur frequently given our model, giving us a high likelihood or can be very unlikely to occur, giving us a low likelihood. Likelihood is important for determining the odds of our data, we use our calculated noise likelihood to re-rank glitch SNRs where $\widetilde{\rho}$ is our re-ranked SNR, ρ is our original glitch SNR, and L is the likelihood ratio of our glitches.

$$\widetilde{\rho} = \sqrt{\rho^2 - 2ln\mathcal{L}} \tag{2}$$

When the likelihood we calculate is ln(L) >1, we should see some improvement for our background compared to the new glitch SNRs. When the likelihood we calculate is ln(L) $\leq$ 0, it results in our background becoming worse as these likelihoods have less impact at higher SNRs and causes glitches to increase in SNR when re-ranked. When we know the likelihood odds of our data we can further use them to determine the ranking statistic and calculate the FAR as an output. For this investigation, we wanted to compare the likelihood of astrophysical vs random noise at a given SNR. We began by simulating astrophysical and random noise. We then normalized the data and checked that they carried the same SNR model and calculated them against each other, as pictured in Figure 2, revealing the SNRs that are more likely to contain astrophysical or random noise. As our SNR increases, we begin to see our astrophysical noise is more likely to occur.

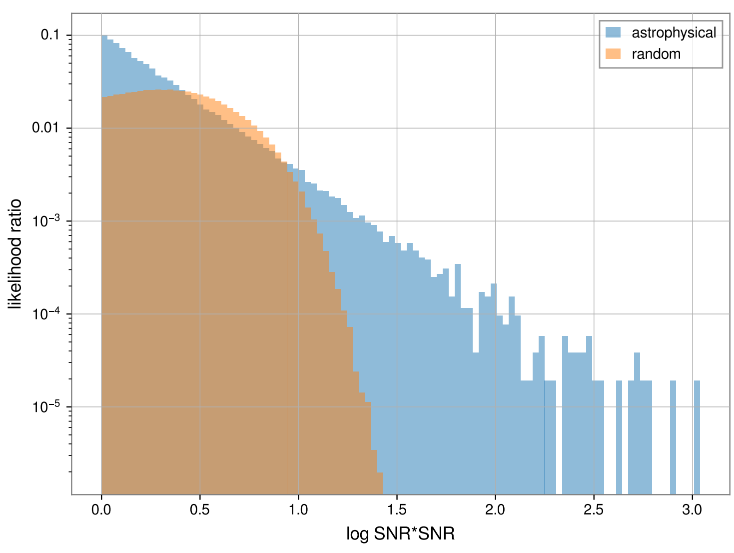

Figure 2: In the plot we are able to see areas where our astrophysical data which is shown in blue or random noise data which is shown in orange has more likelihood of occurring depending on the SNR. We are able to see that above log SNR^2 $\approx$ 1.2 our likelihood ratio is in favor of being astrophysical. In between log SNR^2 0.5 and 1.0 we see that our random noise has a higher likelihood.

4. Current work

With the understanding of how our data's likelihood and noise background changes with flags, we continued onto analyzing a larger data set of glitches from the second observing run (O2) in which we applied flags and analyzed how the glitches ranking statistic and background are affected when flagged sections are removed completely or re-ranked using likelihood. In our current investigations, we take a look at a segment of data from both LIGO detectors (H1, L1) during O2. We begin by filtering our data to look at SNRs above 6.25 and applied a simple χ^2 consistency test to the data. We then generated simulated background data between the two detectors which is seen in Figure 3. Next we produced a second time slide with the data that had a CAT2 flag (flags that correspond to some physical coupling) applied to each data set and removed glitches that fell into the flags. The DQ flags removed 149 of the 799 times. When re-ranking flagged glitches instead of removing them completely, we keep all of the 799 times but still see a change in background.

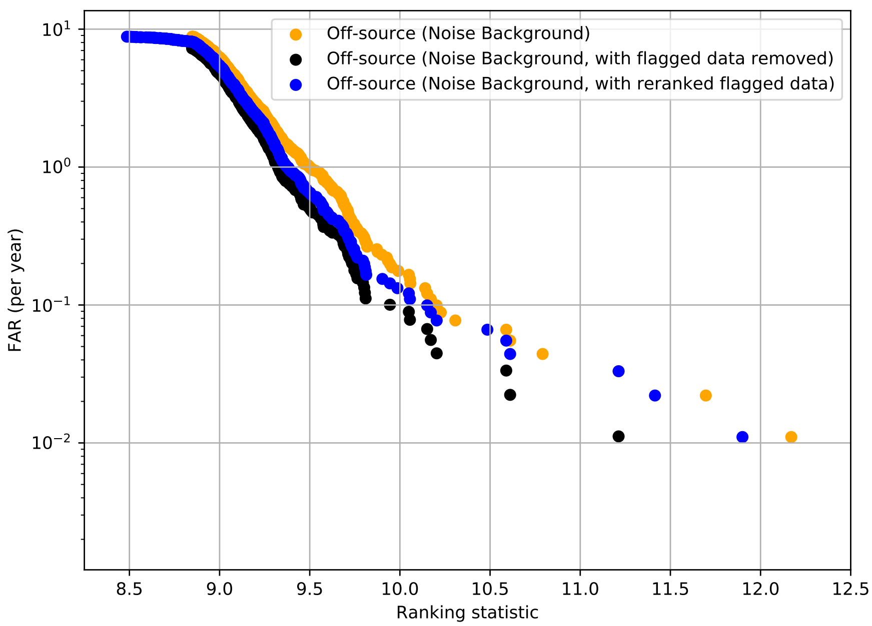

Figure 3: In the plot we can see how the glitches change in the background when applying different veto methods. In the orange we can see the SNRs and FAR of the original glitches. In the black we have our glitches with flagged data removed, which resulted in less data points because they fell with the flags that were applied. This method gave us a VT ratio of 1.27. This led to our background improving but if there was a possible signal in our flag and we removed it, it could affect the search negatively. So, instead of removing the flagged times completely we re-ranked them instead using their likelihood. In the blue we have our glitches with flagged data re-ranked, this method also helped our background improve significantly giving us a VT ratio of 1.07. From this plot we can see our re-rank method works well in reducing the ranking statistic of glitches in the data.

After applying our flagged glitches removal method, we get the ratio of distance to be 1.09, and ratio of time to be 0.99. When applying our flagged glitches re-ranked method, we get the ratio of distance to be 1.02 and a ratio of time to be 1.00. When comparing the ratios of our flags removed versus flags re-ranked methods, we get a ratio of distance to be 0.94, ratio of time to be 1.01 and VT ratio of 0.84. Though there is almost no change between the flags removed and re-rank methods in ratio of distance, ratio of time and ratio of VT, the re-rank method will help us lower the ranking statistic of glitches and make loud signals more significant instead of possibly removing them.

5. Future work

As these methods are simplified prototype versions of tests that PyCBC can run we plan to further investigate tools that could be implemented to improve our statistics such as Gravity Spy [15], Hveto [16], and iDQ [17] to develop data quality flags and vetoes.

6. 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-v2.

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] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, M. Agathos, and et al. Gw190814: Gravitational waves from the coalescence of a 23 solar mass black hole with a 2.6 solar mass compact object. *The Astrophysical Journal*, 896(2):L44, Jun 2020.

[7] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton. FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries. *Phys. Rev. D*, 85:122006, 2012.

[8] Bruce Allen. A chi**2 time-frequency discriminator for gravitational wave detection. *Phys. Rev. D*, 71:062001, 2005.

[9] Alexander H. Nitz, Thomas Dent, Tito Dal Canton, Stephen Fairhurst, and Duncan A. Brown. Detecting binary compact-object mergers with gravitational waves: Understanding and Improving the sensitivity of the PyCBC search. *Astrophys. J.*, 849(2):118, 2017.

[10] Tito Dal Canton et al. Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors. *Phys. Rev. D*, 90(8):082004, 2014.

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

[12] Alexander H. Nitz, Tito Dal Canton, Derek Davis, and Steven Reyes. Rapid detection of gravitational waves from compact binary mergers with PyCBC Live. *Phys. Rev. D*, 98(2):024050, 2018.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.