Effects of Different Data Quality Veto Methods in the PyCBC Search for Compact Binary Coalescences

LIGO Caltech SURF Program 2020, Mentor: Dr. 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 the search. 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 sensitivity of the search for gravitational waves from compact binary coalescences.

1. Introduction

The Advanced Laser Interferometer Gravitational-Wave Observatory (aLIGO) [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). With the increase in sensitivity multiple confident GW detections have been announced so far [3, 4, 5, 6].

The PyCBC [7, 8, 9, 10, 11, 12] search pipeline has been used since the first detection made by aLIGO [13]. PyCBC identifies GW events that are produced by compact binary coalescences (CBCs), which are when two compact objects such as black holes or neutron stars coalesce and experience an inspiral, merger, and ringdown. PyCBC uses matched filtering to compare signals against waveform templates that model what a GW event should look like and distinguishes triggers and potential signals. With the amount of triggers that are identified using matched filtering, loud, short duration glitches are found to sneak across data quality tests. This results in a decrease in search sensitivity and ringing of the match filter [11, 14]. As detections become more frequent, the quality and confidence of detections need to increase. To increase confidence in the search for GWs veto methods need to target glitches that can greatly affect the analysis of a gravitational wave search pipeline such as PyCBC.

In this investigation two different veto methods are analyzed. One veto method will remove triggers and the other will re-rank triggers from data that are coincident with 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. The follow up of how the probability and distribution of triggers change with respect to time will highlight how the veto methods adjusted the triggers to improve PyCBC. The new veto method will also analyze how the PyCBC search responds to different configurations and DQ flags. This paper presents results using O2 data [3] which dates from April 14, 2017 to April 23, 2017 and investigates how triggers are affected with previous and new data quality veto methods by comparing the change in background and sensitivity of the search. In Section 2 the PyCBC pipeline is described in more detail and the issues PyCBC faces with its' current DQ veto method are highlighted. In Section 3 the type of data quality flags used in this prototype are described. In Section 4 the methods used to calculate the likelihood ratios and how the ratios vary with different triggers for the prototype are explained. Section 5 outlines how the new veto method completely removes the possibility of removing a signal by the method of re-ranking triggers. In Section 6 the results of the different veto methods are explained. The results of the new veto method in this investigation can be implemented in the production PyCBC search pipeline in future LIGO observing runs as a new data quality tool to improve the search for gravitational waves from compact binary coalescences.

2. The PyCBC search pipeline

The PyCBC search pipeline determines how significant an event is as compared to noise in the detectors. PyCBC uses modelled CBC templates to identify potential GW events in matched filtering to match signals against waveform templates and re-weighs the relationship with the estimated power spectral density (PSD) of the detectors involved. PyCBC also uses a time slides method, described in Section 2.3, to create a background distribution and measure the false alarm rate (FAR) of recorded events [11]. The detection statistic for triggers and signals is represented by Equation 1 where ρ^2 is the ranking statistic of triggers, $\log p^s(\vec{\theta})$ is the signal distribution, and $\log p^n(\vec{\theta})$ is the noise distribution.

$$\rho^2 \propto 2\left[\log p^s(\vec{\theta}) - \log p^n(\vec{\theta})\right] + constant. \tag{1}$$

2.1. Matched filtering

In order for PyCBC to search for CBCs with confidence, matched filtering uses a template bank which holds a library of GW templates that are made of modelled CBC waveforms. Matched filtering calculates the signal-to-noise ratio (SNR) of the templates and PyCBC then uses a χ^2 signal consistency test to filter the SNRs and remove triggers that do not match the templates very well. The matched filter is expressed by Equation 2 below where $\tilde{s}$ is the data, h represents the template, and S is the PSD.

$$x(t_0) = 4\mathcal{R}\int_0^\infty \frac{\tilde{s}(f)[\tilde{h}^*_{template}(f)]_{t_0=0}}{S_n(f)} e^{2\pi i f t_0} df. \tag{2}$$

2.2. Chi-square method used to calculate a new ranking statistic

The (χ^2) signal consistency test weighs the SNR of triggers identified by the matched filtering. The output of this test is a new ranking statistic based on the re-weighted SNR. In Equation 3 the χ^2 is given where the frequency bins (p) in each template each have their own SNR (ρ_i) and equal amount of power in the template calculated by the matched filter. Here ρ^2_{cos} and ρ^2_{sin} are the SNRs of the matched filter [11]. In Equation 4 larger values of χ^2 indicate a higher likelihood of a noise instead of a signal. For signals, the reduced chi-squared is χ^2_r = χ^2/(2p - 2) [11].

$$\chi^2 = p\sum_{i=1}^{p}\left[\left(\frac{\rho^2_{cos}}{p} - \rho^2_{cos,i}\right)^2 + \left(\frac{\rho^2_{sin}}{p} - \rho^2_{sin,i}\right)^2\right], \tag{3}$$

$$\hat{\rho} = \begin{cases} \rho \Big/ [(1 + (\chi^2_r)^3)/2]^{\frac{1}{6}}, & \text{if } \chi^2_r > 1, \\ \rho, & \text{if } \chi^2_r \le 1. \end{cases} \tag{4}$$

2.3. Time slides

PyCBC's time slides method is used to generate simulated background data by sliding data from one detector against data of the other, in this case the Hanford (H1) and Livingston (L1) detectors. The background distribution generated is useful in determining the statistical significance, also known as the FAR, of triggers. A significant FAR helps determine the likelihood of seeing a detection again with the same network SNR. If the FAR is low the less chance our detection is due to terrestrial noise [14].

3. Data quality flags

Data quality flags are produced to correlate transient noise with disturbances in or around the LIGO detectors. The type of data quality flags used for these disturbances are classified as "category 2 flags" (CAT2), these flags are correlated with some type of physical coupling [15, 14]. The systems that monitor these disturbances are called witness channels which contain sensors such as microphones, accelerometers, and seismometers [16]. Before being used to generate a DQ veto, scientists need to ensure witness channels are sensitive enough to capture noise and not mistake a signal for a glitch. Scientists at the detectors run injections that simulate a GW to test whether or not these systems react to signals efficiently [15]. If a witness channel picks up an injected signal, the channel would not be considered a good witness. When the system is cleared to be a witness channel, it is used to produce DQ vetoes. DQ vetoes produced by DQ flags indicate times that contain problematic noise which can be removed from analysis. Figure 1 shows have an example of a witness channel that produced a DQ veto and efficiently captured a glitch due to noise from a thunderclap. Current veto methods in the production PyCBC utilize DQ flags to veto problematic noise in the data that

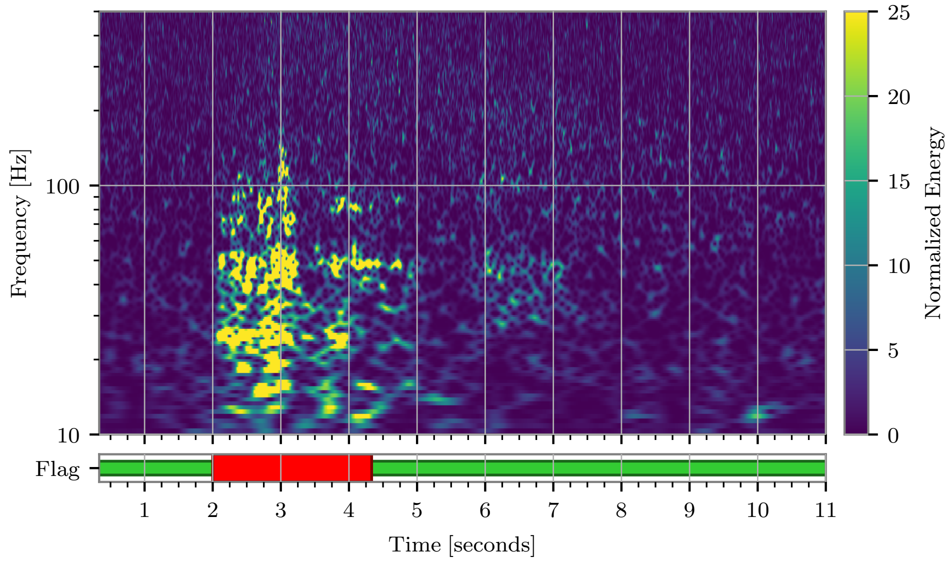

Figure 1: This plot shows an example of an effective data quality flag. This flag was produced from a witness sensor at the corner station containing a microphone. The glitch shown is produced from a thunderclap near the Livingston detector. The color bar at the bottom of the plot shows that the flag became active throughout the duration of the glitch.

make it difficult to run an analysis on potential signals. The current veto method PyCBC runs has experienced problems with removing too much data with DQ flags which has caused the removal of a signal when a glitch is too close or on top of the signal. When ineffective flags are used loud glitches can be missed, especially short duration glitches in the data we look at, causing a decrease in sensitivity. This especially hurts low mass templates, which do not benefit from DQ products.

4. Calculating likelihood ratios to measure a new ranking statistic

Though the χ^2 signal consistency test does a significant job at down ranking triggers with the matched filter, when combined with DQ flags they can effectively target a wider range of problematic triggers than on their own. For this prototype χ^2 and DQ flags are combined to produce a new distribution of triggers that fall inside the flags. This distribution is represented by a likelihood ratio. A likelihood ratio is given by Equation 5.

$$\mathcal{L}(\theta \mid x) = p_\theta(x) = P_\theta(X = x). \tag{5}$$

Here θ is the model or expectation of the 'likelihood function' and x is the measured outcome of the random variable X. The probability of the value x of X for the parameter value θ is written as $P(X = x \mid \theta)$ or $P(X = x; \theta)$. The likelihood is equal to the probability that an outcome x is observed within the parameter model θ and it is equal to a probability density over the outcome x, not over the parameter model θ. If x is very likely in a given model it has a high likelihood. If x is very unlikely to happen it will have a low likelihood. For the new veto method the likelihood ratios of the triggers are calculated inside and outside of vetoed time as seen in Equation 6. This displays how much more likely a trigger would show up during a flagged (or vetoed) time vs all time.

$$\mathcal{L}(t|\theta) = \frac{\mathcal{L}(flag\,time)}{\mathcal{L}(total\,time)}. \tag{6}$$

Equation 6 shows us the trigger rate for vetoed time versus all time by taking the likelihood of seeing a flagged trigger and the likelihood of seeing a trigger in the entire data set. Equation 7 and Equation 8 show how the two individual trigger rate likelihoods are calculated.

$$\mathcal{L}(flag\,time) = \frac{triggers\ flagged}{flagged\,duration} * \frac{1}{triggers\,total}, \tag{7}$$

$$\mathcal{L}(total\,time) = \frac{triggers\,total}{total\,duration} * \frac{1}{triggers\,total}. \tag{8}$$

As template parameters can vary with triggers, they can be accounted for when calculating the likelihood ratios by using templates such as the chirp masses. In Figure 2 a visualization of how likelihood ratios of triggers change as their chirp mass changes is shown.

$$\mathcal{L}(flag) = \frac{\mathcal{L}(flag\,time)}{\mathcal{L}(total\,time)}. \tag{9}$$

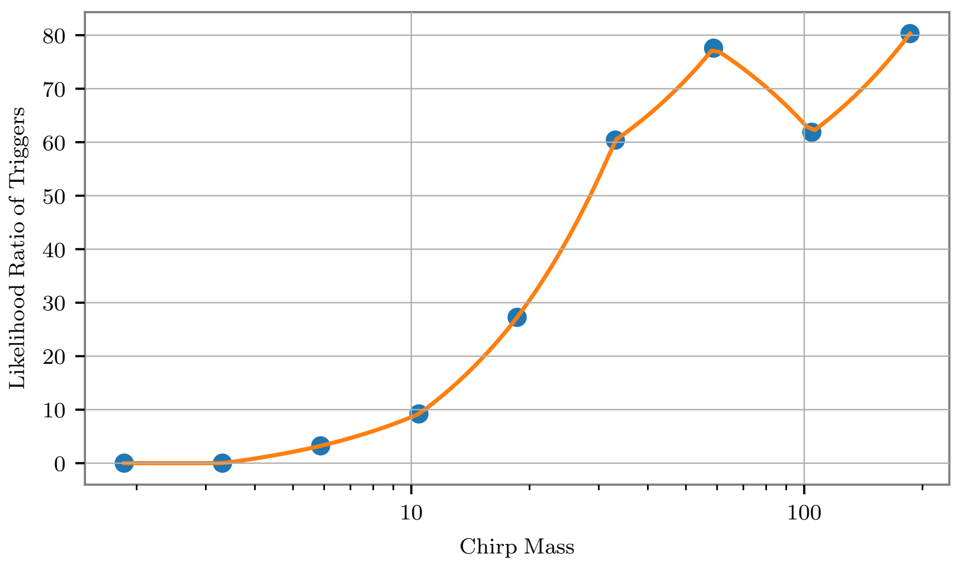

Figure 2: Likelihood ratio of triggers as compared to their chirp mass. The plot shows us as chirp mass increases the likelihood ratio of seeing a trigger inside a flagged time with that chirp mass also increases. Knowing the different likelihoods across the template will be useful in re-ranking triggers more efficiently.

5. Re-ranking triggers

To re-rank flagged triggers the likelihood ratio(s) calculated previously are plugged into Equation 10 where $\widetilde{\rho}$ is the new ranking statistic, ρ is the original ranking statistic, and $\mathcal{L}$ is the likelihood ratio(s) of a data quality flag [17] .

$$\widetilde{\rho} = \sqrt{\rho^2 - 2\log\mathcal{L}(flag)}. \tag{10}$$

The likelihood ratio in Equation 10 should be greater than or equal to 1 because of the natural logarithm in the equation. If the likelihood ratio is equal to 1 there will be no change in the noise ranking statistic after calculation. If the likelihood ratio returns a negative value it would increase the ranking statistic of triggers which would impact the search negatively. If the likelihood were to be zero an error would be returned due to the logarithm. When the likelihood calculations are applied to Equation 9 the original trigger ranking statistics would experience a significant down rank as seen in Figure 3.

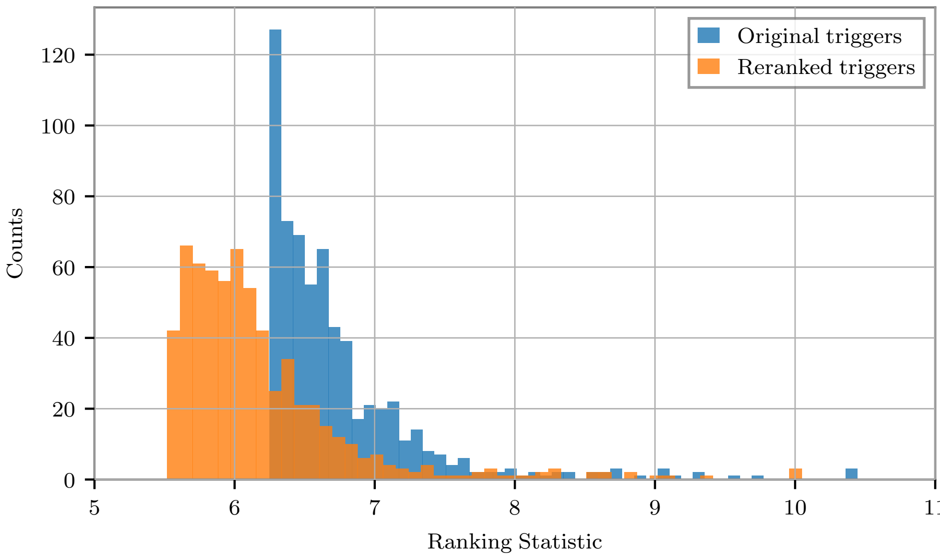

Figure 3: The plot shows how triggers are ranked before and after applying re-ranking. The blue histogram shows the original triggers highlighted during matched filtering, the orange histogram shows the same triggers after applying the new veto method. Using the method described in Section 5 the range of the triggers' detection statistic moves from 6.25 through 10.4 to 5.5 through 10. Also, there is a redistribution of the triggers' counts to group more towards the lower statistics.

6. Results

6.1. Comparing search backgrounds

In the analysis of the different veto methods it is assumed that all injected signals are perfectly recovered and their performance is based off of the background distribution. When the old method and new method are applied to the data the two are compared against the original background to see how effective they are. In Figure 4 the original noise background has triggers with a ranking statistic ranging between 8.8 and 12.3. When the original veto method PyCBC runs is applied there is a change in the triggers' FAR, a removal of some triggers, and the same range of ranking statistics between 8.8 and 11.3 remain. When the new veto method is applied there is also a significant change in the triggers' FAR, but now there is no removal of triggers, and the ranking statistics of triggers range between 8.3 and 11.8. In Table 1 the VT ratios for the original background, old veto method, and new veto method are shown. Looking at the values the VT ratios for both the old and new method are pretty similar and the new method makes up in time for loss in distance. Overall the VT is the same. Since there was no removal of any originally vetoed data this means the new veto method can catch far more signals and this method will always be a positively effective one no matter the flag used.

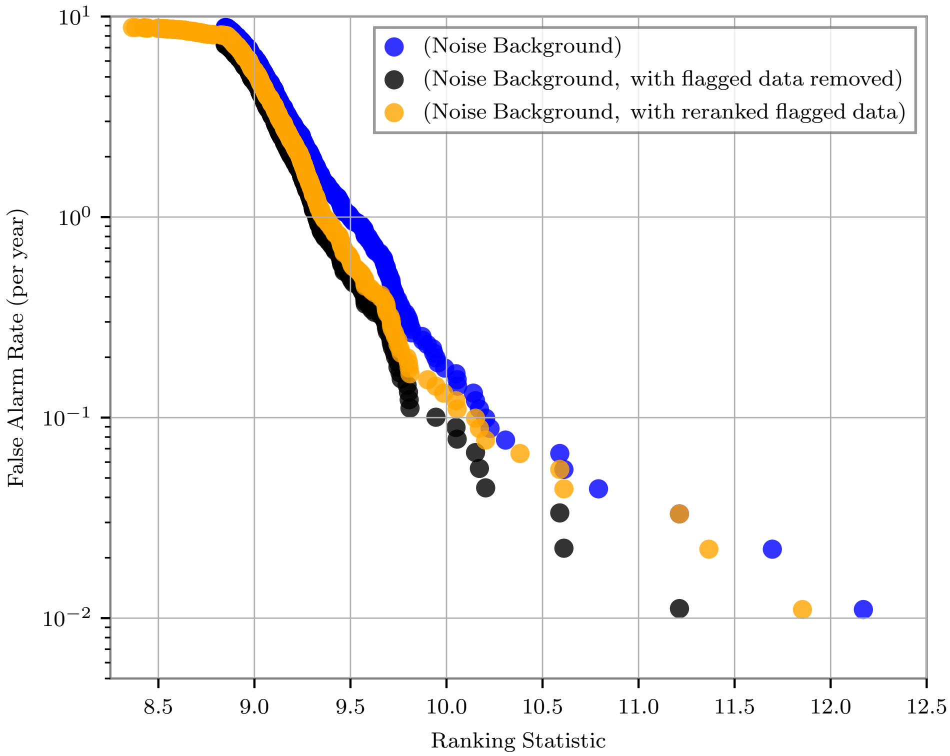

Figure 4: The original ranking statistics for the glitches is represented by the blue curve. The orange curve indicates how glitches were effected with the new method, and the black curve indicates the original method of removing flagged data. The larger groups of glitches highlight how the new method did really well in matching what the original PyCBC veto method did in terms of improving the background and glitch significance.

7. Conclusions

The new veto method re-ranks flagged data instead of removing flagged data completely, assuring no signals can be accidentally removed. The new method increases the significance of signals and reduces the amount of time lost due to flags. With this an increase in the overall number of detectable signals is expected.

As these methods are simplified prototype versions of tests that PyCBC can run further investigations include implementing tools that could improve our statistics. A few extensions to this prototype include expanding on the amount of flags applied, expanding to different DQ products such as Gravity Spy [18], Hveto [19], and iDQ [17] to develop data quality flags and vetoes, and expanding to the updated PyCBC ranking statistic.

Original data vs vetoed:	Ratio of distance:	1.02
	Ratio of time:	0.99
	Ratio of volume $\times$ time:	1.04
Original data vs re-ranked:	Ratio of distance:	1.01
	Ratio of time:	1.00
	Ratio of volume $\times$ time:	1.04
Vetoed vs re-ranked:	Ratio of distance:	1.00
	Ratio of time:	1.01
	Ratio of volume $\times$ time:	1.00

Table 1: In the table the volume and time ratios for the original data, the old veto methods, and the new veto are all compared to each other. The first row shows the comparison between the original data and the old veto method. The second row shows the comparison between the original data and the new veto method. The last row shows the comparison of the old veto method and new veto method. These ratios are based off the background distribution.

8. Acknowledgments

I would like to give a special thank you to Dr. Derek Davis for being an amazing mentor throughout this project, Dr. Alan Weinstein for endless support and enthusiasm, the National Science Foundation for funding this project, the Caltech LIGO SURF program, and LIGO Laboratory. 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. The LIGO SURF program is supported by NSF award PHY-1852081. This work carries LIGO Document number T2000349.

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 $\sim 3.4M_{\odot}$. *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. et al. Abbott. 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] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. 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] T. 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] S. A. Usman et al. The PyCBC search for gravitational waves from compact binary coalescence. *Class. Quant. Grav.*, 33(21):215004, 2016.

[12] A. H. Nitz, T. Dal Canton, D. Davis, and S. 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] B. P. et al. Abbott. Characterization of transient noise in advanced ligo relevant to gravitational wave signal GW150914. *Classical and Quantum Gravity*, 33(13):134001, Jun 2016.
[16] A. Effler, R.M.S. Schofield, V.V. Frolov, G. González, K. Kawabe, J.R. Smith, J. Birch, and R. McCarthy. Environmental Influences on the LIGO Gravitational Wave Detectors during the 6th Science Run. *Class. Quant. Grav.*, 32(3):035017, 2015.
[17] R. Essick, P. Godwin, C. Hanna, L. Blackburn, and E. Katsavounidis. iDQ: Statistical Inference of Non-Gaussian Noise with Auxiliary Degrees of Freedom in Gravitational-Wave Detectors. 5 2020. arXiv:2005.12761.
[18] M. Zevin et al. Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science. *Class. Quant. Grav.*, 34(6):064003, 2017.
[19] J. R. et al. Smith. A Hierarchical method for vetoing noise transients in gravitational-wave detectors. *Class. Quant. Grav.*, 28:235005, 2011.