Digging deeper: finding sub-threshold compact binary merger events in LIGO data

LIGO SURF 2019 Program, Interim Report 2, August 2019

Sierra N. Garza[1,2],
Jonah Kanner[2], **Alan Weinstein**[2], **and Liting Xiao**[2]

[1]California State Polytechnic University, Pomona, CA 91768, USA
[2]LIGO, California Institute of Technology, Pasadena, CA 91125, USA

The LIGO and Virgo detectors have collected gravitational wave (GW) data from three separate observation runs since 2015, with the third run presently collecting data. There have been 10 signals from binary black hole mergers and one binary neutron star merger detected from the first two observation runs and many more from the third run. These detections were all confirmed due to high confidence in their signal-to-noise ratio (SNR); however, there are likely many more unconfirmed signals in the data with lower SNRs. A limitation in the SNR criteria arises when accidental coincidence of "loud" glitches or other rare noise fluctuations in the LIGO detectors can result in high SNRs but are not the product of real GWs. We hope to improve the detection or rejection of sub-threshold events with lower SNRs by computing the Bayesian coherence ratio (BCR): the odds between the hypothesis that the data comprise either a coherent compact-binary-coalescence signal in Gaussian noise or incoherent instrumental features, using parameter estimation. We present a BCR analysis done on Observation Run 3 (O3) event and background data. Initial results provide confidence that the BCR can distinguish between signal and incoherent noise given appropriate parameters, indicating potential to improve sub-threshold event detections.

Gravitational Waves | LIGO | Bayesian Inference | Bayesian coherence ratio | Bilby|Bilby pipe | Observation 3 | Parameter estimation | Singal to noise ratio | Python

Introduction / Background. Initial PyCBC [1]. and GstLAL [2]. pipeline searches have flagged potential GW events as triggers based upon matched-filtering and threshold values [3], Since LIGO-Virgo's search sensitivity scales with binary black hole mergers primary component mass, approximately $VM_1^{2.1}$, higher mass mergers have been detected out to larger distances [4]. However, higher mass binaries merge more quickly and at lower frequencies than lower mass binaries. This is why high mass binaries are harder to detect in the current optimal LIGO band of 10 - 1000 Hz, in which loud glitches tend to fake GWs from higher mass systems. Current compact-binary-coalescence (CBC) searches do not adequately differentiate GWs of lower confidence from detector noise. Bayesian model comparison of coherence may be a way to effectively discriminate whether a marginal trigger, a flagged event just below the SNR from the pipeline searches, is likely a GW signal or instrumental noise from the detectors.

Coherence requires that the strain signals in multiple instruments share a phase evolution consistent with a single astrophysical source, represent a well-described CBC waveform, and be temporally coincident [5]. Instrumental noise transients (glitches) are not expected to fully meet these requirements, whereas GWs are; therefore, allowing the distinction to be made.

The data from these signals contain valuable untapped information to understanding gravitational waves and the characteristics from merging black hole and neutron star systems in the distant universe. Being able to collect and analyze the data from the weaker signals will help fill in the gaps of our understanding of CBC populations.

Objectives / Methods. I will be working on analyzing whether the Bayesian coherence ratio (BCR) can be a reliable method to improve confidence in the signal of marginal triggers and/or improve the rejections of inherent noise in the data in conjunction with current pipeline searches.

BCR is a part of Bayesian inference which is derived from Bayes' theorem. This statistical theorem uses pre-existing information and continually updates predictions to determine conditional probability. Bayes' theorem equation is defined as:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

where A and B are outcomes, P(A|B) is the conditional probability that outcome A occurs given that outcome B has already occurred and is known as the posterior. P(B|A) has the same meaning but with the roles of A and B reversed) and is known as the likelihood. P(A) and P(B) are the marginal probabilities of outcome A and outcome B occurring respectively [6]. P(A) is known as the prior or marginal probability. The prior is how Bayes' theorem allows the use of pre-existing information to determine conditional probability.

We will make use of Bayes' theorem in the following form:

$$P\big(\Theta|data\big) = \frac{P\big(data|\Theta\big) \times P\big(\Theta\big)}{P\big(data\big)}$$

The quantity on the left of the equals sign represents the posterior probability distribution for the parameters (collectively referred to as Θ) of the model describing the data. So, if we're trying to estimate the parameter values of a Gaussian distribution then Θ represents both the mean, μ and the standard deviation, σ (written mathematically as Θ = μ, σ). We identify 15 source parameters that govern a CBC: component masses and spin vectors, right ascention, declination, luminosity distance, orbital inclination, polarization angle, time and phase of coalescence. Instead of outcome B, we'll see data or y = y_1, y_2, ..., y_n[6]. In our case, the data are the time-series of strain values recorded by the detector near the time identified by the search pipelines as containing a candidate signal.

The prior represents the distribution of our belief in the true parameter values. The posterior is also a distribution that represents our belief about the parameter values given the observed data. For the case of data $d = s+n$ consisting of a CBC signal $s(\Theta)$ in Gaussian noise n, the likelihood takes the form of a Gaussian in $n = d-s$.
The denominator, P(data), is the integral of the numerator over all possible values of the parameters Θ. It serves to normalize the posterior probability distribution, and is referred to as the *evidence* for the model representing the data.

The BCR will be computed using different models: GW signal, Gaussian noise, and incoherent glitch. To determine which model is favored in the data, an odds ratio can be used for model selection. The odds ratio is a way to quantify the association between two events, or models in our case. For example, if we computed the odds ratio of the evidence for the GW signal model to Gaussian noise model:

$$OddsRatio = \frac{p(data|GWsignal)}{p(data|GaussianNoise)}$$

If the ratio resulted in a value greater than 1, then the data favors the GW signal model and indicates the data are likely a GW signal. If the ratio value is less than 1, then the Gaussian noise model is favored, indicating the data are likely noise.

For understanding the BCR, I found it useful to understand this following detailed explanation by Max Isi et al. when discussing coherence vs incoherence in [5].

The BCR is the odds between the hypothesis that the data comprise a coherent CBC signal in Gaussian noise (H_S), and the hypothesis that they instead comprise incoherent instrumental features (H_I) - meaning each detector has either a glitch in Gaussian noise (H_G), or pure Gaussian noise (H_N). When using multiple detectors, the BCR equation is written as:

$$BCR \equiv \frac{\alpha Z^S}{\prod_{i=1}^{D}\left[\beta Z_i^G + \Big(1-\beta\Big) Z_i^N\right]}$$

where Z^S is the evidence for H_S, and Z_i^G and Z_i^N are, respectively, the evidences for H_i^G and H_i^N in the i_{th} detector. The arbitrary weights α and β parametrize our prior belief in each model: $\alpha = P(H_S)/P(H_I)$ and $\beta = P(H_G|H_I) = 1 - P(H_N|H_I)$. These priors will be chosen to minimize overlap between the signal and noise trigger populations [5].

Evidences (marginalized likelihoods) are the conditional probability (P) of observing some data (d_i, for detector i) given some hypothesis (H), integrated over all of the parameters associated with that hypothesis (model). For the coherent-signal hypothesis the evidence is

$$\big(Z^S\big) \equiv P\big(\{d_i\}_{i=1}^{D}|H_S\big)$$
$$= \int p\big(\vec{\theta}|H_S\big)p\big(\{d_i\}_{i=1}^{D}|\vec{\theta}, H_S\big)d\vec{\theta}$$

The vector $\vec{\theta}$ represents a point in the space of parameters that describe the CBC signal, such as the component masses and spins; the terms in the integrand are the prior, $p(\vec{\theta}|H_S)$, and the multi-detector likelihood, $p\big(\{d_i\}_{i=1}^{D}|\vec{\theta}, H_S\big) = \prod_{i=1}^{D} p(d_i|\vec{\theta}, H_S)$. The specific functional form of the single-detector likelihood, $p(d_i|\vec{\theta})$, is derived from the statistical properties of the noise (e.g. a normal distribution for a Gaussian process). The integral is performed numerically using algorithms like nested sampling. In our case, the data d_i are the calibrated Fourier-domain output of each detector but could generally be any sufficient statistic produced from it [5].

Because of their inherently unpredictable nature, it is impossible to produce a template that a priori captures all features of a glitch. Therefore, we define a surrogate glitch hypothesis by the presence of simultaneous, but incoherent, CBC-like signals in different detectors. Thus, for the i_{th} detector, the glitch evidence is

$$\big(Z_i^G\big) \equiv P\big(d_i|H_G\big)$$
$$= \int p\big(\vec{\theta}_i|H_G\big)p\big(d_i|\vec{\theta}_i, H_G\big)d\vec{\theta}_i$$

where now we allow for a different set of signal parameters $\vec{\theta}_i$ at each detector. We will set the priors $p(\vec{\theta}_i|H_G) = p(\vec{\theta}_i|H_S)$ and the likelihood $P(d_i|\vec{\theta}_i, H_G) = P(d_i|\vec{\theta}_i, H_S)$, but this may be relaxed to better capture specific glitch features, if necessary. The surrogate H_G model captures the portion of glitches that lie within the manifold of CBC signals and, in a sense, corresponds to the worst possible glitch – one that looks exactly like coincident CBC signals. Variations of this strategy have been used before in the analysis of compact binary coalescences, minimally modeled transients, and continuous waves. Other searches also make use of likelihood ratios in the detection process, but they do not rely on signal coherence [5].

Finally, because we assume a perfect measurement of the detector noise power-spectral-density (PSD), the Gaussian-noise evidence is just the usual null likelihood. For our Fourier-domain data, this is:

$$(Z_i^N) \equiv P(d_i|H_N) = N(d_i)$$

Where N(d_i) is the product of a multidimensional normal distribution with zero mean and variance derived from the noise PSD. In principle, this could be generalized to marginalize over poorly known PSD parameters if needed [5].

We want to calculate the BCRs for triggers produced in all observing runs to detect weak GW signals and potentially define empirical probability distributions that would allow us to obtain likelihood ratios to use for trigger classification.

The BCRs will be computed with the likelihood values calculated from running Bilby [7] jobs on the LIGO cluster. We will run hundreds of Bilby jobs on injected signals, glitches, confirmed signals, and marginal triggers analyzed with an astrophysical and respective noise models for both Advanced LIGO (aLIGO) detectors.

We will compute and apply the BCR to O2 (Observation Run 2) and O3 background triggers in effort to reject any glitches. This will allow an updated false alarm rate (FAR) to be used for O3 event analysis. A BCR < 1 would allow for the rejection of that trigger as a GW event because it would favor the odds of the hypothesis that the data is comprised of incoherent instrumental noise.

Various plots will be produced to visualize the results from using the BCR to compare real and simulated signals to inherent noise. I will utilize Thomas Alford's Bilby tutorial python code [8] to produce plots in Equation 1, following the discussion in [5].

Progress. For this project I am developing a thorough understanding of Bayesian inference and parameter estimation and its potential applications to GW research. I have also become familiar with Alford's python code [8]; which has allowed me to understand how to utilize LIGO data and set up to run multiple Bilby jobs on LIGO's computer cluster.

I investigated if there are any overlapping trigger times from several published sub-threshold trigger catalogs, such as GWTC-1 [3], 1-OGC [9], two from Princeton [10, 11]. I extracted the intervals of data that surrounded the trigger times listed in the various catalogs and write corresponding hdf5 files. This was done after first converting all published trigger times into GPS time, the time in seconds from January 06, 1980. For each catalog's list of triggers, I appended them to a list and sorted them from lowest to highest trigger times to discern if there were any overlapping triggers within a catalog. To make this determination, I compared whether the ending time of the first trigger was greater or equal to the starting time of the following trigger. I used a loop to iterate through each list. I then compared multiple catalogs for overlapping trigger times by appending the sub-lists into one list and resorting it so that the triggers were ordered lowest to highest GPS time. By running the new list through the same loop, I was able to determine no overlapping trigger times from the four sub-threshold catalogs, given a specified time range of ± 0.1 s surrounding the published trigger times. However, I am expanding my specified time range to identify any sub-threshold triggers that are reasonably close in time, given that multiple sub-threshold catalogs utilized different pipeline searches to produce their respective catalogs.

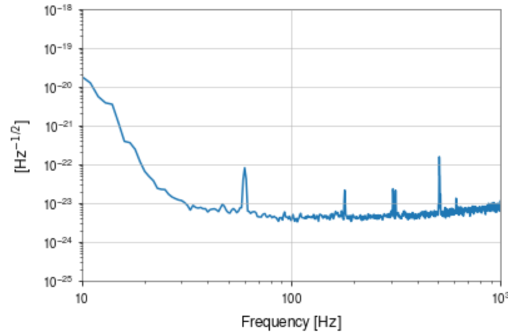

Fig. 1. Frequency plot of the amplitude spectral density (ASD) over a GPS time frame (1242442965.45, 1242442969.45). This data was taken from the channel L1:GDS-CALIB _STRAIN_CLEAN. Frequency range specified for LIGO-Virgo sensitivity band. Below 10 Hz is not properly calibrated. Above 10^3 Hz is affected by detector quantum noise.

In python, I have produced frequency plots that have been averaged over a stretch of time (Figure 1) for various trigger times to visualize how the amplitude spectral density (ASD) in the data evolves over specified time intervals.

I have used a computed set of BCR and SNR data files to compute the log_{10}BCR to produce density plots against corresponding SNR distributions with varying α and β values, as shown in Figure 2. Contour and scatter plots show how the foreground, background, and selected GW trigger relate log_{10}BCR vs SNR. The relation between these data subsets are also shown in a histogram with counts vs log_{10}BCR, and cumulative distribution function vs log_{10}BCR to represent the survival function from the selection of background triggers and foreground signals [5]. These visualizations help indicate whether the α and β values need to be adjusted and whether the BCR is effectively distinguishing GW signals from inherent noise [8].

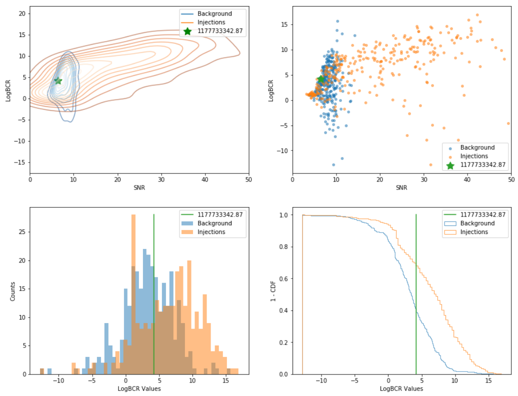

Fig. 2. Density Plots with $\alpha = 3.62804 \times 10^{-11}$ and $\beta = 3.26975 \times 10^{-06}$. Adjustments in α and β are needed due to no clear distinction between the background and injections in the data.

Analysis. To submit and run bilby_pipe jobs on the LIGO computer cluster I need to have several files for each job. An .ini file contains a set of user input arguments that is used to initialize a bilby_pipe job. These arguments include the out directory for the results, detectors, channels, gps trigger time, duration, sampler type, and prior file or gps and time shift files. The detectors and channels are specified so that bilby can locate the data for the duration around the specified gps trigger time. The sampler type I used is dynesty, which is a dynamic nested sampling algorithm that performs parameter estimation and evidence calculation. The prior file is a text file that contains ranges of the 15 parameters that characterize a specified CBC. The gps file is a text file that contains the gps trigger time to be used in the computation. The time shift file is another text file containing the respective time shifts to be applied to Hanford (H1) and the Livingston (L1) detectors for each gps time.

I cloned Avi Vajpeyi's GitLab repository of Bilby_Pipe code [12] to analyze his results of computing lnBCR's for time shifted data of GW150914 and hacking his code to analyze other O3 super event triggers, background, and confirmed O1 and O2 GW events. Bilby_Pipe is a Python 3 tool for automating the process of running bilby for gravitational parameter estimation on computer clusters [13].

Since aLIGO does not operate continuously and its observation run has only spanned several months, time shifting the data allows us to artificially generate more data containing accidental coincidence of noise events in two or more detectors to compare trigger times and calculate FARs and lnBCRs. Time shifting the data is when a specified time difference is applied to one or both detector gps times. This results in a new simulated data set that can be reanalyzed for trigger events such as glitches. When computing lnBCRs I generally used time shifts of subtracting 1000 seconds to the initial trigger time and adding 1000 seconds to both detector gps times. I expected a similar lnBCR as when no time shift was applied to the data because the time shifts should cancel each other out, my results reflected this. I also applied a time shift of adding 500 seconds to the initial trigger time and the Livingston detector (L1) gps time while subtracting 500 seconds to the Hanford detector (H1) gps time. The deliberate time shifting of the detectors in opposite directions is expected to represent just noise in the data and therefore resulting in a small or negative lnBCR value, which initial results seem to reflect.

I was able to produce similar lnBCR results as [12] for the time shifted GW150914 data, with values ranging 11 - 14. This gave me confidence that I could submit bilby jobs to condor and produce results. I then applied the same time shifts to the loudest 2 O3 background triggers determined by the Coherent Wave Burst (CWB) group of an Intermediate Mass Black Hole (IMBH) search, with H1 gps times of 1241670026 and 1239849040.59, respectively as reported by [14]. I wanted to test the code on only two background triggers because I wanted to determine the proper signage for the time shift difference between the H1 and L1 gps times. I chose to keep H1 as my baseline and apply the additional time shift to L1. By subtracting L1 gps time from H1 gps time, I got a value in seconds. For the loudest background trigger, I chose to keep the same sign as the computed value. For the second loudest trigger, I applied a sign change to that value and used it in the time shift code. Knowing that these triggers were background and I was using a BBH prior, I expected negative lnBCR values. For the first trigger, I did get lnBCR values ranging between -13 and -11. For the second trigger, I got much smaller lnBCR values between -6 and +1. This indicated to me that the correct signage for the additional time shifts would be whichever sign the value was when I subtracted L1 from H1 gps time. I corrected the signage for the second trigger and got resulting lnBCR values between -14 and -12. The large negative values represent that the glitch model is supported more than the signal model, which gives us confidence that we can likely reject this trigger as a CBC event.

I am working on a python 3 script that will be able to generate and submit bilby_pipe jobs for the top 100 loudest background triggers from the CWB group. I have successfully read in the text file containing the relevant H1 and L1 gps times for each trigger. I have isolated the top 100 trigger times in a list and have computed the time differences between H1 and L1 for each trigger in a separate list. I am working on generating the appropriate files to run time shift data and compute lnBCR values to compare with my other results.

I then wanted to calculate the lnBCR values for all 22 of the O3 super events posted to the public alerts page on GraceDB as of August 01, 2019 [15]. To accomplish this, I needed to change some information for each job that I wanted to submit to the cluster. For the .ini file I needed to change the label and out directory to include each event appropriate name, for example S190521g . I had to insert the corresponding gps trigger time as listed by the preferred event on GraceDB. I also had to change the channel to GDS-

CALIB_STRAIN_CLEAN, which corresponded to O3 data. I found the available channels through the terminal with the command:

FrChannels /archive/frames/O3/hoft/H1/H-H1_HOFT_C00-12390/H-H1_HOFT_C00 1239085056-4096.gwf

O3 represents the third observation run of LIGO. hoft is what it sounds like: h(t) = strain in the data. H1 is the Hanford detector and the rest is just specifying the calibration round (C00), gps time (1239085056) and the sampling frequency (4096 Hz). I chose the CALIB_STRAIN_CLEAN channel because it seemed the most applicable data to my lnBCR analysis of the strain.

I first wanted to analyze O3 events using a prior profile that was run on GW150914. This prior had a mass range of 10-80 solar masses and a minimum mass ratio of 0.125. From some initial lnBCR values, I chose to adjust the GW150914 prior by extending the mass range from 8 - 80 solar masses and the minimum mass ratio to 0.100. I chose to make this change because I wanted to avoid false posteriors if component masses were at or near the prior boundaries. I used this same altered prior for the likely BNS super events. However, since they likely contain a neutron star,their component masses are below 3 solar masses and I do not expect this prior to produce positive lnBCR values.

I wanted to also run an analysis of the O3 super events using an IMBH prior. Since an IMBH event would be comprised of large component masses I decided to use an IMBH prior generated by Vajpeyi [12]. This prior has a mass range for 70 – 150 solar masses and a minimum mass ratio of 0.08.

After my first couple rounds of submissions for the O3 super events I was outputting lnBCR values that were smaller than I thought they would be. Due to some investigation, it was found that when extracting the evidence to produce the results of the lnBCR values, the α and β values were both set to 1.

When I was working with Alford's IMBH_Data_Analysis python code [8], it was clear that $\alpha = 1$ and $\beta = 1$ are not good values to separate background from injection signals.

Instead, in that same notebook and in [5], it was found that appropriate values are $\alpha = 10^{-6}$ and $\beta = 10^{-4}$. When I changed these values and re-ran the super events the lnBCR values were more aligned with what we expected.

To further analyze these results, I looked over intrinsic corner plots, as seen in Figure 4. The corner plot displays the probability distribution between the 15 CBC parameters and shows the likely value as a result of this parameter estimation. I also looked at the checkpoint trace plots (Figure 5) which show the progress of the parameter estimation dynamic nested sampling algorithm. Ideally, we want to see a nice Gaussian distribution in these plots to indicate that our prior parameter ranges, such as minimum and maximum masses, are adequate for the data we want to analyze.

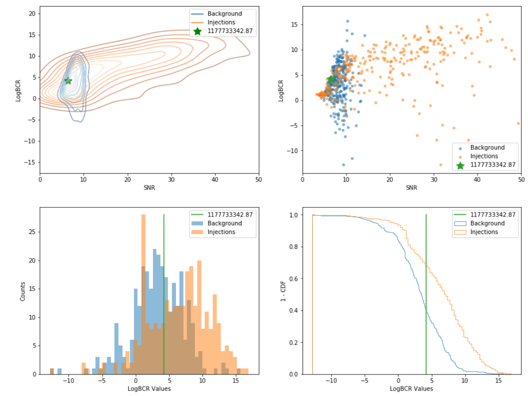

Fig. 3. Density Plots. α = 1 and β =1. The resulting plots show that there is no clear differentiation between background triggers and injections in the data.

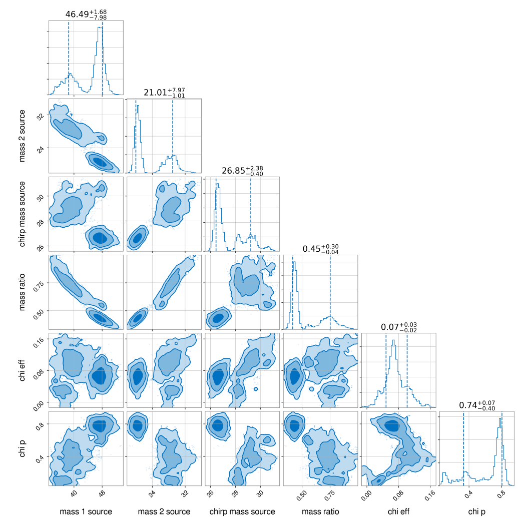

Fig. 4. Intrinsic corner plot for GW150914. The top two rows and two left columns show the probability distributions for the two component masses, in solar masses. The likely values and their uncertainties are displayed above the histogram.

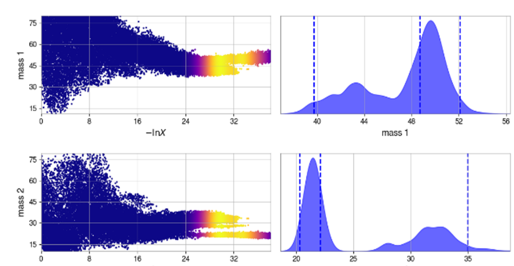

Fig. 5. Checkpoint trace plots for GW150914. The two rows show data for the two component masses, in solar masses. The left plot shows the convergence of the likely component mass values as a result from the parameter estimation. The right column shows the resulting probability distribution in reference to the prior boundaries.

Future Work. Moving forward, I would like to rerun the 4 likely BNS super events with a prior that is structured for such a CBC signal and compare lnBCR values. I would also like to submit bilby_pipe jobs for at least the top 100 loudest background triggers for O3 [14]. For those jobs, I am planning on using an IMBH prior since these triggers were flagged by an IMBH pipeline search. I would also like to run a series of injections in the O3 data and compute lnBCR values. Having lnBCR values from O3 super events, background, and injections would provide me with a nice sampling to suggest whether computing the lnBCR values would improve the detection of marginal and sub-threshold compact binary merger events in LIGO data. If these results are promising, I would also like to calculate the lnBCR values for all sub-threshold trigger events found in the GWTC-1, 1-OGC, and both Princeton catalogs from O1 and O2. I would also compute lnBCR values on O1 and O2 background, GW events, and injections. This should provide a comprehensive analysis of the lnBCR computation and its potential to enhance GW detections by updating the FAR as a result of identifying any likely GW events and/or rejecting any likely triggers due to instrumental noise. An improved FAR can be used to improve our confidence in GW detection events in the current and future LIGO data.

Bibliography

1. Usman et al. The pycbc search for gravitational waves from compact binary coalescence. 2016.
2. Messick et al. Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data. 2017.
3. 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. 2018.
4. M. Fishbach and D. E. Holz. Where are ligo's big black holes? *Astrophys. J. Lett.*, 851(2), 2017.
5. Isi et al. Enhancing confidence in the detection of gravitational waves from compact binaries using signal coherence. 2018.
6. J. Brooks-Bartlett. Probability concepts explained: Bayesian inference for parameter estimation. Available at https://towardsdatascience.com/probability-concepts-explained-bayesian-inference-for-parameter-estimation-90e8930e5348,, 2019.
7. ashton2019bilby. Bilby: A user-friendly bayesian inference library for gravitational-wave astronomy. 2019.
8. Thomas Alford. Bilby_tutorial. 2019. Available at https://git.ligo.org/thomas.alford/bilby_tutorial/,.
9. E. Thrane and C. Talbot. An introduction to Bayesian inference in gravitational-wave astronomy: parameter estimation, model selection, and hierarchical models. 2019.
10. Venumadhav et al. New binary black hole mergers in the second observing run of advanced ligo and advanced virgo. 2019.
11. Venumadhav et al. A new search pipeline for compact binary mergers: Results for binary black holes in the first observing run of advanced ligo. 2019.
12. Avi Vajpeyi. Bilby_pipe. 2019. Available at https://git.ligo.org/avi.vajpeyi/bilby_pipe/tree/master/,.
13. bilby_pipe. Available at https://lscsoft.docs.ligo.org/bilby_pipe/,, 2018.
14. Background non zero lag - network l1-h1 [16.0-2048.0]. Available at https://ldasjobs.ligo.caltech.edu/~vedovato/reports/O3_LH_IMBHB_ONLINE_BKG_run1/postprod/M1.C_rho1gt5.C_bin1_cut_nodq.R_rMRA_i0cc00_i1rho0_freq16_2048/,, 2019.
15. Ligo/virgo public alerts. *GraceDB As of Aug*, 2019. Available at https://gracedb.ligo.org/superevents/public/O3/,.