Spring 5-2016

An Improved Pipeline to Search for Gravitational Waves from Compact Binary Coalescence

Samantha Usman

An Improved Pipeline to Search for Gravitational Waves from Compact Binary Coalescence

A Capstone Project Submitted in Partial Fulfillment of the Requirements of the Renée Crown University Honors Program at Syracuse University

Samantha Anne Usman

Candidate for Bachelor of Science
and Renée Crown University Honors
May 2016

Honors Capstone Project in Physics

Capstone Project Advisor: _____
Duncan Brown
Professor of Physics

Capstone Project Reader: _____
Peter Saulson
Professor of Physics

Honors Director: _____
Stephen Kuusisto, Director

Acknowledgements:

I’d like to thank my collaborators: Alexander H. Nitz, Ian W. Harry, Christopher M. Biwer, Miriam Cabero, Collin D. Capano, Tito Dal Canton, Thomas Dent, Stephen Fairhurst, Marcel S. Kehl, Drew Keppel, Badri Krishnan, Amber Lenon, Andrew Lundgren, Alex B. Nielsen, Larne P. Pekowsky, Harald P. Pfeiffer, Matthew West and Joshua L. Willis. I’d like to thank Peter Saulson, my collaborator and capstone reader. I’d like to thank my parents for supporting me through my research. I’d especially like to thank my advisor Duncan Brown who pushed me when I wanted to give up.

Abstract:

We describe the search pipeline, called *PyCBC*, used to identify gravitational wave candidates in Advanced LIGO's O1 search. We give an overall summary of the pipeline including a general layout and the methods for calculating the signal-to-noise ratio and the chi-squared statistic. We then give detailed explanations of removing noise transient before processing, called *gating*; the way to create a bank of template waveforms and the process of matched filtering these waveforms against the data; the chi-squared statistic's role in eliminating noise transients; the latest coincidence testing method, which requires candidates to be both nearly simultaneous and have the exact same parameters; and the method of translating a candidate's significance into a probability of being a true signal. We compare this pipeline to the one used in Initial LIGO's fifth and sixth science runs and report a 40% increase in sensitive volume with no significant change to computational cost.

Executive Summary:

Gravitational-wave physics is at the forefront of scientific exploration. Einstein's theory of general relativity treats gravity as the warping of spacetime by the presence of mass. When mass accelerates, it creates ripples in spacetime. On September 14, 2015, and almost 100 years after Einstein first predicted their existence, the Laser Interferometer Gravitational-Wave Observatory (LIGO) directly observed the gravitational waves from a binary black hole merger, verifying a key prediction of general relativity. My capstone thesis describes the development and testing of a new search pipeline that was used to measure the significance of LIGO's first detection. The discovery of the binary black hole source GW150914 is reported in "Observation of Gravitational Waves from a Binary Black Hole Merger" published in Physical Review Letters **116**, 06110 2 (2016). Additional details of the search are described in the paper "GW150914: First results from the search for binary black hole coalescence with Advanced LIGO" which has been submitted to Physical Review D.

Black holes are objects so massive that they warp spacetime into a singularity preventing even light from escaping their gravitational field. Gravitational waves are ripples in spacetime which carry information about changing gravitational fields out to far reaches of the universe. As two black holes or neutron stars orbit around each other, their gravitational fields warp spacetime. Information about the changing gravitational field of the source is carried away as gravitational waves. The goal of gravitational-wave astronomy is to use the information carried by gravitational waves to study the universe, just as we have used the information carried by light to study the universe for hundreds of years.

LIGO uses laser interferometry to detect minute changes in distances caused by gravitational waves passing through the Earth. The two LIGO detectors, which are located in Hanford, Washington and Livingston, Louisiana, are also sensitive to vibrations from the environment that cause noise in the detectors. The three main noise sources are seismic motion from the ground, thermal vibrations from within the detector, and noise due to the quantum nature of light itself. These noise sources can be a problem because the detectors are very sensitive, even sensitive enough to pick up the vibrations from waves crashing against the seashore hundreds of miles away. The expected amplitude of typical astrophysical gravitational-wave sources will be comparable to the amplitude of the detector noise. Because of this, noise sources can hide the signals of gravitational waves. To detect signals, LIGO has designed a sophisticated *search pipeline* that can find gravitational waves buried in detector background noise. The purpose of a search pipeline is to identify gravitational-wave signals in the output of the detector network and to measure the significance of observed sources.

The shape of the gravitational waves from binary black holes varies greatly depending on the masses of the black holes and how fast they are rotating. Since we do not know exactly what the black holes look like in advance, the pipeline uses hundreds of thousands of different gravitational waveforms and compares each of them to the output of the detectors. These waveforms can last from a fraction of a second to several minutes. By using an operation called the *matched filter*, the pipeline checks every template waveform against every second of LIGO data to see if the data could contain any of the possible wave types and the *signal-to-noise ratio* (*SNR*) that describes how likely it is that any given piece of data contains a signal. If the matched filter returns a high number for a particular waveform at a certain time in the data, this time is

flagged as a *trigger* and is considered to be a gravitational-wave candidate. The value of a trigger's SNR is a measure of how loud the gravitational-wave candidate is.

Loud noise sources in the detector may fool the matched filter by generating triggers with large SNR values, even if they are not due to gravitational waves. If the data contains a genuine gravitational wave, the power of the data should be distributed in the same way as the power of the theoretical waveform. The pipeline uses a χ^2 *test* to compare the amount of power distributed in the detector data to the power distribution expected from the waveform. Noise transients, do not have the same power distribution and will return a large χ^2 value. The trigger's SNR is divided by the value of the χ^2 test to construct a *re-weighted SNR*. The reweighted SNR reduces the loudness of triggers that look like noise compared to triggers that look more like signals.

The theory of General Relativity predicts that gravitational waves travel at the speed of light and the Hanford and Livingston detectors are approximately 3000 kilometers apart. This means that gravitational waves should arrive at the detectors with a time difference of no more than 10 milliseconds. The pipeline thus performs a *coincidence test* to reduce false identification of signals by requiring that a signal is seen in both detectors within the gravitational-wave travel time between the observatories. Signals that are see in coincidence as considered to be *candidate events* that could be gravitational waves. To measure the overall significance of a these candidate events, the pipeline computes the square root of the combined squares of the re-weighted SNR in each detector. This quantity is known as the detection statistic. The larger its value, the more likely a candidate is to be a gravitational wave.

However, the value of the detection statistic does not give us the whole story. It does not tell us the probability that a given event is real. To do this, the pipeline compares the re-weighted

SNR of gravitational-wave candidates to that of background noise from the detector in order to determine their significance. Since we cannot turn off our gravitational wave sources and there is no theoretical model for the noise in the detectors, we must use *time shifts* to estimate the background data. In a time shift, data from one detector (and all of its resulting gravitational wave candidates and noise triggers) are shifted in time with respect to the other detector's data. The pipeline then performs another round of coincidence testing. Any resulting coincidences cannot be gravitational waves, since they did not occur simultaneously in real time. By doing many of these time shifts, we can artificially create thousands of years' worth of background data with just a few weeks of LIGO data. This allows us to estimate the *false-alarm rate* of the search as a function of the detection-statistic value; this is the rate at which noise produces candidate events with a given detection-statistic value. Finally, the candidate events are compared to this set of background noise and are determined to be statistically significant or not.

After describing the new search pipeline, we present a comparison with the pipeline used in previous LIGO searches and found that the new pipeline was more sensitive, with no change to the cost. We test the pipeline's sensitivity by using *injections* which are fake signals that are artificially added to the detectors' data. By checking to see when the pipeline can pick out these gravitational waves from the detector data, we can estimate to what volume of space the pipeline is sensitive. My analysis showed that the new pipeline was 40% more sensitive to low-mass binary systems (like binary neutron stars), with no significant change to the computational cost or the sensitivity to high-mass systems. This pipeline was used to identify and confirm LIGO's first gravitational-wave detection, GW150914.

Contents

1. Introduction

The detection of the binary black hole merger GW150914 by the Laser Interferometer Gravitational-wave Observatory (LIGO) has established the field of gravitational-wave astronomy [1]. LIGO [2, 3] will be joined by the VIRGO [4, 5] and KAGRA [6] detectors in the near future, forming an international network of gravitational-wave observatories. Beyond the expected regular detections of binary black holes [7], compact-object binaries containing a neutron star and a black hole or two neutron stars are likely candidates for detection by this network in the coming years [8, 9].

In this paper, we describe a new search pipeline to detect gravitational waves from compact-object binaries. The purpose of a search pipeline is to identify gravitational-wave signals in the output of the detector network and to measure the significance of observed sources. The pipeline described here, called *PyCBC*, searches for gravitational waves from compact binary coalescences [10]. The methods used directly descend from the pipelines [11–15] used to search for compact-object binary coalescence with the the first-generation LIGO and Virgo detectors [16–24]. The pipeline has been re-written to incorporate new algorithms that improve the sensitivity of the search and that reduce its computational cost. We provide a complete description of the search pipeline, emphasizing the new developments that have been made for the advanced-detector era. To demonstrate the use of the pipeline, we show that by re-analyzing data from LIGO's sixth science run that the new pipeline can achieve a $\sim 40\%$ increase in sensitive volume to binary neutron stars when compared to the pipeline used in the last LIGO-Virgo science run [24]. The results of using this pipeline in Advanced LIGO's search for binary black holes are presented in Ref [1] and described in detail in Ref. [25].

This paper is organized as follows: Section 2 provides an overview of the new search pipeline and the methods used to detect gravitational waves from compact-object binaries in LIGO data. Section 3 gives a description of the new developments implemented in this pipeline. Section 4 compares the performance of the new pipeline to that of the pipeline used in the sixth LIGO science run and Virgo's second and third science runs [24]. Finally, Section 5 summarizes our findings and suggests directions for future improvements.

2. Search Pipeline Overview

The purpose of a search pipeline is to identify candidate signals in the detector noise and provide a measure of their statistical significance. The amplitude of many gravitational-wave sources will be comparable to the amplitude of the detector noise. Even for loud sources, the statistical significance of candidate detections must be empirically measured, since it is not possible to shield the detectors from gravitational-wave sources and no theoretical model of the detector noise

exists. If the detector noise contained only a stationary and Gaussian component, matched filtering would be the optimal method of detecting these signals since the gravitational-waveforms of compact-object binaries can be modeled using a combination of analytical and numerical methods [26]. Futhermore, in this case the the matched filter signal-to-noise ratio (SNR) would suffice as a detection statistic since its distribution in stationary, Gaussian noise is well known [27]. In practice, the detector data contains non-stationary noise and non-Gaussian noise transients and Figure 1 shows the additional steps that must be taken to mitigate false detections and to assign a statistical significance to candidate signals. In this section, we provide an overview of the methods used in the search pipeline.

In addition to possible signals, the calibrated strain signal from the detectors [28, 29] contains two classes of noise: (i) a primarily stationary, Gaussian noise component from fundamental processes such as thermal noise, quantum noise, and seismic noise coupling into the detector; and (ii) non-Gaussian noise transients of instrumental and environmental origin [30]. To eliminate the worst periods of detector performance, data quality investigations [31, 32] characterize detector data into three general classes: (i) the data is suitable for astrophysical searches, (ii) the data is polluted with enough noise that the data should be discarded without being searched, or (iii) the data can be filtered, but candidate events that lie in intervals of poor data quality should be discarded due to the presence of an instrumental or environmental artifact. Data quality investigations are conducted both independently of the search pipeline (by looking only at instrumental performance) and within the pipeline by looking at the effect of instrumental artifacts on the noise background of the search. These investigations are very effective at reducing the background; however noise transients of unknown origin can still remain after vetoes are applied. In Section 3.1 we describe a new method of removing non-Gaussian noise transients by applying a window to the detector data that zeroes out the data around the time of a noise transient (called *gating*).

Since the waveform of the target signal is well modeled, the pipeline uses matched filtering to search for these signals in detector noise. If the total mass of a compact-object binary is lower than $M \lesssim 12\,M_\odot$ [33, 34] and the angular momenta of the compact objects (their *spin*) is small [35,36] (as is the case for binary neutron stars), then the inspiral phase can be well modeled using the post-Newtonian approximations (see e.g. Ref. [37] for a review). For high-mass and high-spin binaries, analytic models tuned to numerical relativity can provide accurate predictions for gravitational waves from compact binaries [38–42]. Since we do not *a priori* know the parameters of gravitational waves in the data, a *bank* of template waveforms is constructed that spans the astrophysical signal space [43–51]. The template bank is constructed so that the loss in matched-filter SNR due to the discrete nature of the bank is no more than $\sim 3\%$. The exact placement of the templates depends on the detector's noise power spectral density. The pipeline described here places

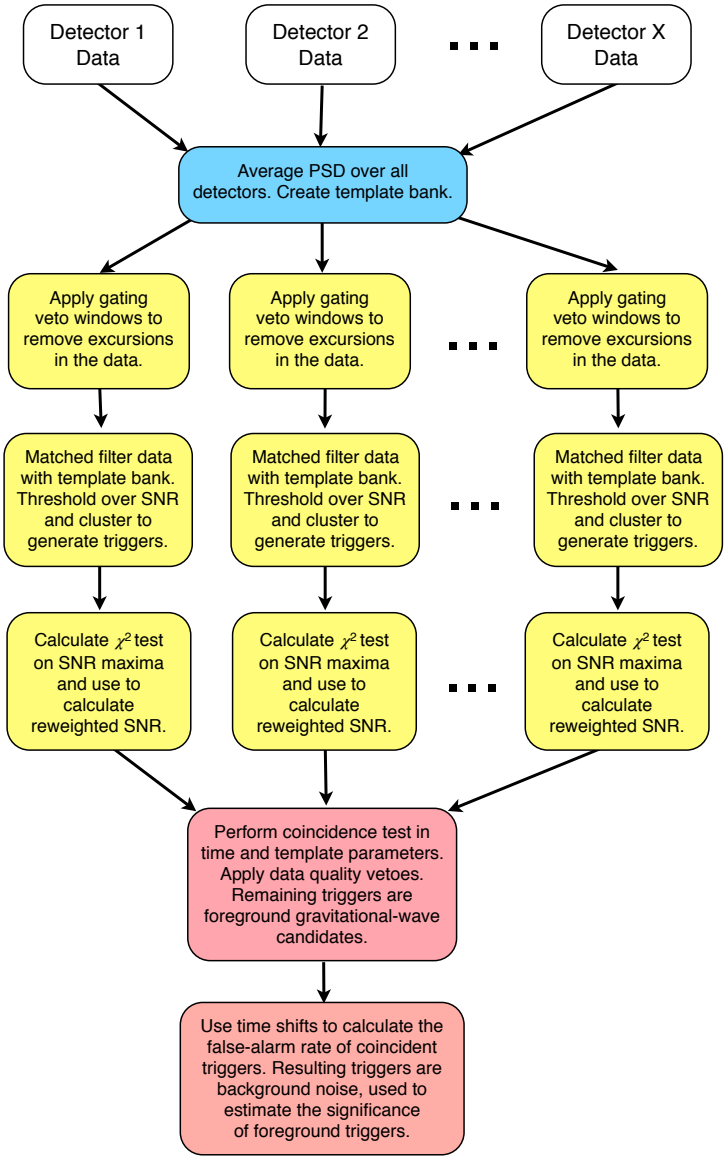

Figure 1. A flowchart indicating the different steps used in the search pipeline. Data from the detectors are used to create an average power spectral density that is used to place a bank of templates that cover the search parameter space (blue box). Times when the detector data contain loud noise transients are removed, or *vetoed*. The data from each detector is then matched filtered and triggers are generated by thresholding and clustering the signal-to-noise ratio time series. A chi-squared test is computed on the SNR maxima and the SNR is re-weighted by the value of the chi-squared statistic to better distinguish between signal and noise (yellow boxes). The pipeline determines which triggers survive time and template coincidence, discarding triggers that lie in times of poor data quality (red box). The triggers that pass the coincidence and data quality tests are labelled candidate events. Finally, multiple time-shifts are used to generate a noise background realization that is used to measure the significance of the candidate events (bottom box).

templates using a single noise power spectral density, averaged over all of the time and all detectors used in the search. The same template bank is used to construct the matched filters for all detectors in the network. We describe the method used to construct this template bank in Section 3.2.

The search computes the matched-filter SNR $\rho(t)$ for each template independently in each detector. The matched filter consists of a weighted inner product in the frequency domain used to construct the (squared) signal-to-noise ratio, given by

$$\rho^2(t) = \frac{(s|h_c)^2}{(h_c|h_c)} + \frac{(s|h_s)^2}{(h_s|h_s)} = \frac{(s|h_c)^2 + (s|h_s)^2}{(h_c|h_c)}, \tag{1}$$

where h_c and h_s are the two orthogonal phases of the template $h(t)$ and have the same normalization, and

$$(s|h)(t) = 4\mathrm{Re}\int_{f_{\mathrm{low}}}^{f_{\mathrm{high}}} \frac{\tilde{s}(f)\tilde{h}^*(f)}{S_n(f)} e^{2\pi i f t}\,\mathrm{d}f. \tag{2}$$

Here $\tilde{s}(f)$ denotes the Fourier-transformed detector data, defined by

$$\tilde{s}(f) = \int_{f_{\mathrm{low}}}^{f_{\mathrm{high}}} s(t) e^{-2\pi i t f} dt, \tag{3}$$

and $\tilde{h}(f)$ denotes the Fourier-transformed template waveform. $S_n(f)$ is the one-sided power spectral density of the detector noise. The frequency limits f_{low} and f_{high} are determined by the bandwidth of the detector's data. The search identifies the times when the matched-filter SNR exceeds 5.5 one of the detector's data set. The pipeline then applies a clustering algorithm, which takes the largest value within a predefined window of $\rho(t)$ and identifies maxima in the SNR time series. This process yields a list of candidate signals, which are called *triggers*. The matched-filtering, thresholding and clustering algorithms used are described in Section 3.3.

Triggers generated by matched filtering the data against the template bank are subject to a chi-squared test that determines if the time-frequency distribution of power in the data is consistent with the expected power in the matching template waveform [13]. To construct this test, the template is split into p frequency bins of equal power, and a matched filter ρ_l constructed for each of these bins. The χ^2 statistic is computed as

$$\chi^2 = p\sum_{l=1}^{p}\left[\left(\frac{\rho_c^2}{p} - \rho_{c,l}^2\right)^2 + \left(\frac{\rho_s^2}{p} - \rho_{s,l}^2\right)^2\right], \tag{4}$$

where ρ_c^2 and ρ_s^2 are the two orthogonal phases of the matched filter. Previous searches fixed the number of bins p. Lower-mass binary systems, such as binary neutron stars, lose energy to gravitational waves more slowly than higher-mass systems. Consequently, the waveforms of lower mass systems are longer, having

more gravitational-wave cycles in the sensitive band of the detector. The pipeline described here allows the number of bins to be specified as a function of the intrinsic parameters of the template. This allows the search to use more bins in the chi-squared test for longer templates, making the test more effective. This pipeline also uses a new computationally efficient algorithm for computing the chi-squared statistic, described in Section 3.4.

Large values of the chi-squared test indicate that a trigger is more likely to be from a noise transient. For true signals, the reduced chi-squared, $\chi^2_r = \chi^2/(2p-2)$, should be near unity. To suppress the matched-filter SNR of triggers caused by noise transients, the matched-filter SNR is re-weighted [15, 52] according to

$$\hat{\rho} = \begin{cases} \rho \Big/ [(1 + (\chi^2_r)^3)/2]^{\frac{1}{6}} , & \text{if } \chi^2_r > 1, \\ \rho, & \text{if } \chi^2_r \leq 1. \end{cases} \tag{5}$$

The search requires that signals are observed with consistent parameters in the detector network, as described in Section 3.5. To be considered a candidate event, triggers must be observed with a time of arrival difference less than or equal to the gravitational-wave travel time between detectors, with an additional window to account for uncertainty in the measurement of the time of arrival. The triggers must also be observed with the same template in both detectors. Triggers that survive the time and parameter coincidence test are referred to as *candidate events*.

The quadrature sum of the reweighted SNR $\hat{\rho}$ in each detector is the detection statistic used to rank the likelihood that a trigger is due to a gravitational-wave signal. To assign a statistical significance to detection candidates, the pipeline measures the false-alarm rate of the search as a function of the detection-statistic value $\hat{\rho}_c$. Since it is not possible to isolate the detectors from gravitational waves, it is impossible to directly measure the detector noise in the absence of signals. This, together with the non-stationary and non-Gaussian nature of the noise, means that the false-alarm rate of the search must be empirically measured. This is done by applying a time-shift δ to the triggers from one detector relative to another. The minimum time offset is chosen to be larger than twice the coincident window. Triggers that survive time-shifted coincidence tests represent coincidences due to noise alone since genuine signals will not be time coincident in the background data set. Many time shifts are used to create a large background data set that is used to measure the background noise and estimate the search's false-alarm rate. The false-alarm rate is then used to assign a p-value to each candidate event as described in Section 3.6.

Different templates in the bank can respond to detector noise in different ways. This means that the search background is not uniform across the template bank. To maximize sensitivity and provide a better estimate of event significance, the search sorts events—both the background and the event candidates—into different classes. The search assigns a p-value to the event candidates by measuring their significance against background events from the same class. To account for having searched

multiple classes whose response is not completely independent [53], this significance is decreased by a trials factor equal to the number of bins, $n_{\rm bins}$. This final value is then used to determine the statistical significance of a candidate event. This process is described in Section 3.6.

3. Pipeline Improvements

In this section we describe the methods and algorithms used in the PyCBC search pipeline, which improves upon the ihope pipeline described in Ref. [15]. Section 3.1 describes a new method for identifying and removing non-Gaussian noise transients from the data prior to filtering. Section 3.2 describes a new method for the construction of template banks and the use of a single template bank in all detectors, and for the entire duration of the search. Section 3.3 describes the improvements made to the matched-filtering used in the new pipeline, including improvements to the detector noise power-spectral density estimation and methods for identifying maxima in the signal-to-noise ratio time series. Section 3.4 describes the new algorithm for constructing the chi-squared signal-based veto used by the pipeline and compares the computational cost of this algorithm to the implementation used in the first-generation detector searches. The substantially reduced cost of our implementation allows us to create a simpler topology which uses one stage of matched filtering as opposed to the two stages used in Ref. [15]. The new topology computes both the matched filter SNR and the chi-squared test before the coincidence step, as shown in Figure 1. Section 3.5 describes the method that the pipeline uses to determine if a signal is observed in coincidence in all the detectors in the network. Section 3.6 describes the method that the pipeline uses to measure the significance of detection candidates.

3.1. Removal of non-Gaussian noise

The input to the search pipeline consists of calibrated gravitational-wave strain data [28], a list of the time intervals during which the detectors are operating nominally (called *science mode*) and additional metadata describing the quality of the detector data [31, 32]. Data-quality investigations examine the strain data and the detectors' environmental and control channels to identify times when detector performance is degraded by noise of instrumental or environmental origin. For periods of unstable detector operation, data can either be discarded from the search before filtering or triggers can be discarded after filtering during the coincidence step. Time intervals that are discarded are known as *veto windows*. However, despite extensive detector characterization investigations, the data still contain non-stationary and non-Gaussian noise which can affect the astrophysical sensitivity of the search. Although loud transients (or *glitches*) that survive data-

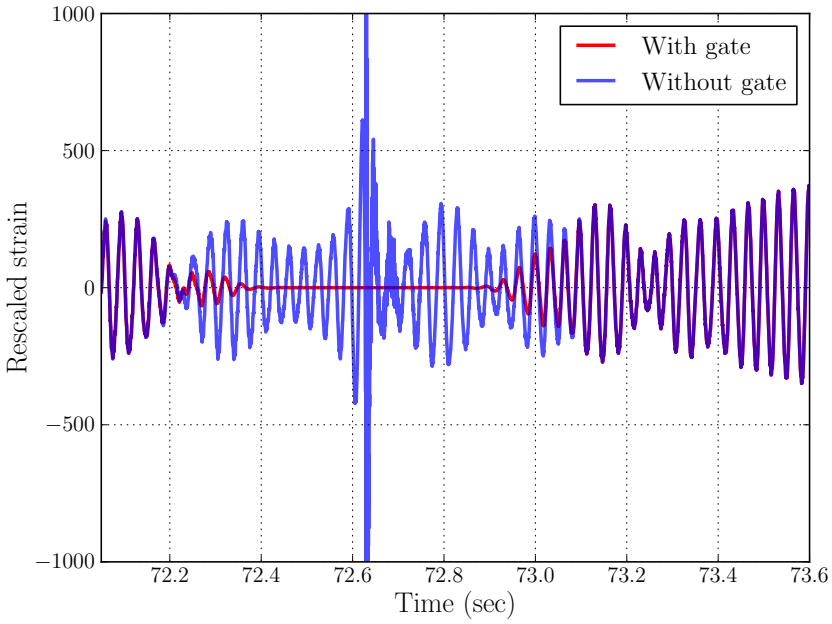

Figure 2. The effect of gating on a loud noise transient. Both lines show strain data just prior to applying the Fourier transform for the frequency-domain matched filter. The y-axis is rescaled so that the strain data around the transient is visible. On this scale, the transient has a peak magnitude over 5,000. The blue line shows the data before applying the Tukey window, the red line shows the data after.

quality investigations are suppressed by the chi-squared test, they can reduce search sensitivity through two mechanisms: *dead time* due to the clustering algorithm and *ringing* of the matched filter; both of these mechanisms are related to the impulse response of the matched filter. In this section, we explain the origin of these features and propose a new method for reducing their effect on the search's sensitivity.

The impulse response of the matched filter is obtained by considering a delta-function glitch in the input data $s(t) = \delta(t-t_g)$, where t_g is the time of the transient. Although not all types of glitches are of this nature [54], many types of loud noise transients can be approximated as $s(t) = n(t)+\delta(t-t_g)$. For example, Figure 2 shows a typical loud transient glitch from LIGO's sixth science run that behaves in this way. For such a glitch, the matched-filter SNR given by Eq. (1) will be dominated by

$$\rho^2(t) \approx I_c^2(t_g - t) + I_s^2(t_g - t), \tag{6}$$

where

$$I_{c,s}(t_g - t) = \int \frac{\tilde{h}_{c,s}(f)}{S_n(f)} e^{2\pi i f(t_g - t)}\, df \tag{7}$$

is the impulse response of each of the two phases of the matched filter. This effect is illustrated in Figure 3 (left) which shows the result of filtering the glitch in Figure 2

through a template bank and generating triggers from the matched-filter SNR time series. The black circles show the triggers generated by the ringing of the filter due to the glitch. The largest SNR values occur close to the time of the glitch and are due to the impulse response of the template h. The shoulders on either side of this are due to the impulse response of the inverse power spectrum $1/S_n(f)$. To ensure that the impulse response of the filter is of finite duration, the inverse power spectral density is truncated to a duration of 16 seconds in the time domain before filtering [14]. However, the length of the template can be on the order of minutes, leading to very long impulse response times for the matched filter. This leads to the two effects described above: an excess of triggers around the time of the glitch which can increase the noise background of the search and a window of time containing multiple noise triggers that can make it difficult to distinguish signal from noise. Although the use of the chi-squared veto to construct the reweighted SNR suppresses the significance of these triggers, it does not completely remove them from the analysis. This is shown in Figure 3 (left) where the blue points show the triggers that remain after calculating the reweighted SNR for each trigger and applying a threshold of $\hat{\rho} > 5.5$. It can be seen that the increased trigger rate around the time of the glitch is still present.

The pipeline described here implements a new method of removing non-stationary transients, called *gating*. Noise transients in the strain data are identified and the data is then zeroed out prior to matched filtering. To zero the data, the input data $s(t)$ is multipled by a window function centered on the time of the peak. A Tukey window is used to smoothly roll the data to zero and prevent discontinuities in the input data. The effect of this gating on the strain data is shown in Figure 2 where the input data is zeroed for 1 s around the time of the glitch, with the window applied for 0.5 s before and after this interval. The effect on the triggers is shown in Figure 3 (right). The blue points show the triggers produced by the search when the windowed data is filtered through the template bank. In addition to removing the loud triggers with SNR ~ 1000 at the time of the glitch, gating removed the additional triggers with SNR ~ 10 that are generated by the ringing of the filter before and after the glitch. This improves search sensitivity by reducing the amount of data corrupted by the glitch to only the windowed-out data, and reduces the overall noise background by removing noise triggers that could possibly form coincident events.

This process requires the identification of the time of noise transients that will be removed by gating. These times can be provided to the pipeline as a list generated by a separate search for excess power in the input strain data (typically with a high SNR threshold) [55, 56]; this method was used in Ref. [1]. Alternatively, a list of times to gate may be constructed internally by the pipeline (called *auto-gating*). To do this, the pipeline measures the power spectral density of the input time series. The data is Fourier transformed, whitened in the frequency domain, and then inverse Fourier transformed to create a whitened time-series. The magnitude of the whitened strain

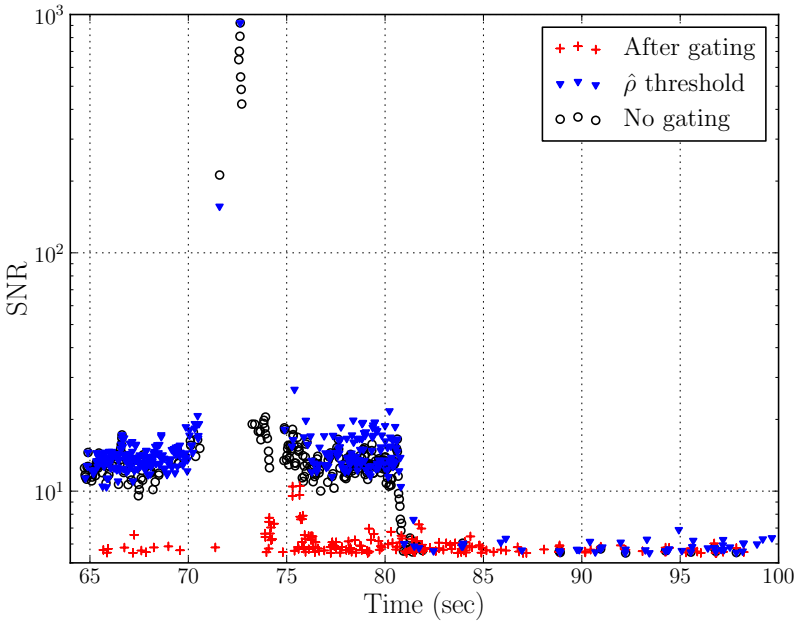

Figure 3. Response of the search to a loud glitch. The matched-filter SNR of the triggers generated is shown as a function of time. Black circles show the triggers generated immediately after filtering without gating applied to the input data. Blue triangles show the triggers that remain after the re-weighted SNR is computed for each trigger and a threshold of $\hat{\rho} > 5.5$ is applied. Although the significance of these triggers is suppressed by the re-weighted SNR, many triggers still remain which can increase the noise background of the search. The red crosses show the triggers produced by the search after gating is applied to the input data. The triggers caused by the glitch in the ± 8 s around the transient have been removed.

data is computed and times exceeding a pre-determined threshold are identified. The clustering algorithm of Ref. [14] is used to identify peaks that lie above the threshold. These times are used as input to the gating algorithm with a window of a pre-determined length applied centered on the peak time. The peak threshold, clustering time window, and gating window length are empirically-tuned parameters that are specified as input to the pipeline.

3.2. Template Bank Placement

The search pipeline takes a template bank as input which is used to search for signals in the target parameter space using matched filtering [43–47]. The template bank is constructed by specifying the boundaries of the target astrophysical space and the desired *minimal match*; this is the fractional loss in matched-filter SNR caused by the discrete nature of the bank. To determine the loss in matched-filter SNR caused by differences in the intrinsic parameters of two waveforms, a metric can be

constructed (either analytically or numerically, depending on the type of template used) that locally measures the fractional loss in matched-filter SNR for varying intrinsic parameters of the templates [46]. This metric (and hence the template placement) depends on the power spectral density of the detector noise. In this section, we describe an improved method for constructing the power spectral density used to place the bank.

Calculating the matched-filter SNR requires an estimate of the detector's noise power spectral denisity $S_n(f)$. The frequency-domain correlation between the signal $\tilde{s}(f)$ and template $\tilde{h}(f)$ is weighted by the inverse of the detector's noise power spectral density; this supresses the correlation in regions where the detector noise is large [26]. The sensitivity of the LIGO and Virgo detectors varies with time, and so the noise power spectral density used in the matched filter must track the variation in the detector's noise. This is achieved by periodically recomputing the noise power spectral density during the observation period. To perform the matched filter, detector data was divided into blocks that have a typical length of 2048 s, with the power spectral density for the filter re-estimated on this timescale.

Previous searches also re-computed the noise power spectral density used to place the template bank on the same time scale; separate template banks were created for each detector, and the bank was re-placed every 2048 s. The pipeline described here allows the use of a single template bank for all detectors and the entire duration of the search. Creating a single, fixed bank for the entire duration of the search requires averaging the detector's noise power spectral density over the full observation period and then using this globally averaged power spectral density to place the template bank. We have explored several different methods to create a single noise power spectral density valid for the duration of the analysis period [57]. We find that the harmonic mean provides the best noise power spectral density estimate for placing the template bank. Specifically, we measure the power spectral density of the noise every 2048 seconds over the observation period using the median method of Ref. [14] to construct N_s power spectra S_n. We then construct the harmonic mean power spectral density defined by averaging each of the separate f_k frequency bins according to

$$S_n^{\mathrm{harmonic}}(f_k) = N_s \Bigg/ \sum_{i=1}^{N_s} \frac{1}{S_n^i(f_k)} \ . \tag{8}$$

The use of the harmonic mean was motivated by Ref. [51] which shows that the harmonic sum of the individual detector power spectral densities in a network yields the same combined signal-to-noise ratio as a coherent analysis of the detector data. We next average the power spectral densities computed for the two detectors to create a single template bank that is used to construct the same template bank for both detectors. In Section 4, we compare the sensititivity and the computational cost of using a single template bank constructed with an averaged power spectral

density and that of using regenerated template banks with shorter power spectral density samples.

3.3. Matched Filtering

A core task of the search pipeline is to correlate the detector data against a bank of template waveforms to construct the matched filter signal-to-noise ratio. The matched filtering used in our search pipeline is based on the FindChirp algorithm developed for use in the Initial LIGO/Virgo searches for gravitational waves from compact binaries [14]. In this section we describe two improvements to the FindChirp algorithm: changes to the noise power spectral density estimation and a new thresholding and clustering algorithm that identifies maxima in the signal-to-noise ratio.

The matched filter in Eq. 2 is typically written in terms of continuous quantities, e.g. $s(t), h(t)$, and $S_n(f)$. In practice, the pipeline works with discretely sampled quantities, e.g. $s_j \equiv s(t_j)$, where s_j represents the value of $s(t)$ at a particular time t_j. Similarly, $S_n(f_k)$ represents the value of the noise power spectral density at the discrete frequency f_k. The input strain data is a discretely sampled quantity with a fixed sampling interval, typically $\Delta t = 1/4096$ s. Fourier transforms are computed using the Fast Fourier Transform (FFT) algorithm in blocks of length $T_B = 256$ s. For this block length, the number of discretely sampled data points in the input time-series data s_j is $N = T_B/\Delta t = 256 \times 4096 = 2^{20}$. The discrete Fourier transform of s_j is given by

$$\tilde{s}_k = \sum_{j=0}^{N-1} s_j e^{-2\pi i j k/N}, \tag{9}$$

where $k = f_k/(N\Delta t)$. This quantity has a frequency resolution given by $\Delta f = 1/(N\Delta t)$.

To construct the integrand of the matched filter, the discrete quantities $\tilde{h}_k$, $\tilde{s}_k$, and $S_n(f_k)$ must all have the same length and frequency resolution. In the FindChirp algorithm, this is enforced by coupling the computation of $\tilde{s}_k$ and $S_n(f_k)$. The power spectral density is computed by averaging (typically) 15 blocks of $\tilde{s}_k$, leading to a large variance in the value of the estimate of the power spectral density. The impulse response of the quantity $1/S_n(f_k)$ when it is computed in this way is the same length as the filter output. To prevent corruption of the filter due to wrap-around of the FFT, the power spectral density must be truncated to a smaller length in the time domain [14, 58]. This truncation length is typically 16 s.

The pipeline described here implements an improved PSD estimation and truncation algorithm. The computation of $S_n(f_k)$ and $\tilde{s}_k$ are decoupled, allowing many more averages to be used to compute $S_n(f_k)$. The pipeline calculates the power spectral density estimate using 16 s time-domain segments. For an input filter length of 2048 s, this means that the power spectral density is computed by averaging 127

blocks, rather than 16, leading to a much lower variance in the estimate. To ensure that the power spectral density has the same resolution as the input data segments $\tilde{s}_k$, the computed $S_n(f)$ is inverse Fourier transformed, zero-padded to the same length as the input s_j and then Fourier transformed into to the frequency domain. This interpolates $S_n(f)$ to the desired resolution. This method of constructing the power spectral density also ensures that the non-zero length of $S_n(f)$ in the time domain is 16 s, and so additional truncation of the inverse power spectrum is not necessary.

Once the matched-filter SNR time series ρ_j^2 has been computed, the final step of the filtering is to generate *triggers*. These are maxima where the signal-to-noise ratio time series exceeds a chosen threshold value. For either signals or noise transients, many sample points in the signal-to-noise ratio time series can exceed the signal-to-noise ratio threshold, and so the pipeline must apply a clustering algorithm to identify maxima. Since a real signal will have a single, narrow peak in the SNR time series, we wish to keep only local maxima of the time series that exceed the threshold.

The algorithm used in this pipeline divides the 256 seconds of signal-to-noise ratio time series produced by each application of the filter into equal 1 s windows. The search then locates the maximum of the time series within each window. As the maximization within each window is independent from other windows, this allows the clustering to be parallelized over the windows for efficiency on multi-core processing units. If the desired clustering window is greater than 1 second, then the resulting list of maxima—potentially one for each window within the segment—is further processed to reduce the number of maxima recorded: the candidate trigger in a window is kept only if it has a higher signal-to-noise ratio than both the window before and after it, ensuring that it is a local maximum in the signal-to-noise ratio time series. This clustering algorithm is not only more computationally efficient, but also produces less dead time than the running maximization over template length described in Ref. [14].

3.4. Signal-Based Vetoes

The chi-squared test introduced in Ref. [59] and developed in Ref. [13] has been shown to be a powerful discriminant between signals and noise in searches for gravitational waves from compact-object binaries. However, it can be computationally intensive to calculate. Since the detection statistic $\hat{\rho}_c$ requires the computation of the reweighted SNR given in Eq. 5, the pipeline must compute the chi-squared statistic for every peak in the matched-filter SNR time series. In this section, we describe a more computationally efficient implementation of the chi-squared veto algorithm.

The search pipeline used in the initial LIGO-Virgo era computed the chi-squared statistic for an entire filter segment. This is computationally expensive, as computing

the matched-filter SNR for each of the p sub-templates increases the cost of the search by a factor of p. Two methods were used in the initial LIGO-Virgo searches to mitigate this cost: (i) the chi-squared test was only computed for an FFT block length of $T_B = 256$ s if any sample point in the matched-filter SNR exceeds the trigger-generation threshold in SNR [14]; (ii) the chi-squared test was only computed for templates that have one or more coincident triggers in an FFT block [15]. While the second method can reduce the cost of computing the chi-squared test, this saving is offset by the fact that the chi-squared test is needed to compute the re-weighted SNR detection statistic for background triggers as well as triggers that are time-coincident. The pipeline described here implements a significant optimization to the chi-squared veto. This improved algorithm calculates the chi-squared test only at the time samples corresponding to clustered triggers in the matched-filter SNR time series. To do this, the pipeline uses an optimized integral over frequency, rather than computing the FFT. If data-quality is poor and the number of triggers in an FFT block is large, the FFT method becomes more efficent. We quantify this cross-over point below and the pipeline falls back to using the FFT method in this circumstance.

To compare the computational cost of the two methods, we consider the calculation of the p matched-filters for the naïve chi-squared test implementation. Computation of the ρ_l^2 for each of the p bins requires an inverse complex FFT. For a data set containing N sample points the number of operations is $p \times 5N\log(N)$. (Since the FFT dominates the operations count, we have neglected lower-order terms that do not significantly contribute to the computational cost.) The pipeline only needs calculate the chi-squared test at the peaks of the matched-filter SNR time series. Since the SNR for any peak is already known, the additional information needed to compute the chi-squared test in Eq. 4 is given by the right hand side of

$$\frac{\chi^2 + \rho^2}{p}[j] = \sum_{l=1}^{p} \rho_l^2[j], \tag{10}$$

where $[j]$ is the set of points in the matched-filter SNR time series where the chi-squared statistic must be computed. The chi-squared statistic can be computed from the kernel of the matched filter $\tilde{s}(f)\tilde{h}(f)/S_n(f)$, which is stored when the matched-filter SNR is computed. We denote the discrete version of this frequency-domain quantity as $\tilde{q}_k$, which contains $N/2+1$ complex numbers [58].

Suppose that the filter, thresholding, and clustering algorithms have identified N_P peaks in the matched-filter SNR time-series where the chi-squared statistic needs to be computed. For the set of times of the peaks $[j]$, we can re-write Eq. 10 as

$$\frac{\chi^2 + \rho^2}{p}[j] = \sum_{l=1}^{p} \left(\sum_{k=k_l^{\min}}^{k_l^{\max}} \tilde{q}_k e^{2\pi i j k/N} \right)^2 , \tag{11}$$

where $k_l^{\min,\max}$ denote the frequency boundaries of the p bins and are given by

$k_l^{\min,\max} = f_l^{\min,\max}/\Delta f$. If $N_P \ll N$, then directly computing the chi-squared test at each of the $[j]$ points is more computationally efficient than computing the $\rho_l^2(t)$ time series, as was done in Ref. [14]. We can further reduce the computational cost of the chi-squared by considering the fact that the exponential term requires the explicit calculation of $e^{-2\pi ijk/N}$ at

$$k_{\max} = \frac{f_{\text{Nyquist}}}{\Delta f} = \frac{1}{2\Delta t \Delta f} = \frac{N}{2} \tag{12}$$

points. This can be reduced to a single computation of the exponential term by pre-calculating $e^{2\pi ij/N}$ once and then iteratively multiplying by this constant to obtain the next term needed in the sum. To do this, we write Eq. 11 in the following form:

$$\frac{\chi^2 + \rho^2}{p}[j] = \sum_{l=1}^{p} \left(\sum_{k=k_l^{min}}^{k_l^{\max}} \tilde{q}_k e^{2\pi ij/N} (e^{2\pi ij/N})^{k-1} \right)^2 . \tag{13}$$

This reduces the computational cost of each term in the sum to two complex multiplications: one to multiply by the pre-computed constant $e^{2\pi ij/N}$ and one for the multiplication by $\tilde{q}$. Computing the right hand side of Eq. 13 then requires the addition of two complex numbers for each term in the sum. The total computational cost to compute the chi-squared test for N_P points is then $14k_{\max} \times N_P = 7N \times N_P$.

For small numbers of matched-filter SNR threshold crossings, this new algorithm can be significantly less costly than calculating the chi-squared statistic using the FFT method. However, if the number of threshold crossings N_P is large, then the FFT method will be more efficient due to the $\log N$ term. The crossover point can be estimated for p chi-squared bins as

$$N_p = \frac{p \times 5N \log(N)}{14 k_{\max}} = \frac{5}{7} p \log(N), \tag{14}$$

although this equation is approximate because the computational cost of an FFT is highly influenced by its memory access pattern. For a typical LIGO search where $N = 2^{20}$, the new algorithm is more efficient when the number of points at which the χ^2 statistic must be evaluated is $N_p \lesssim 100$. For real LIGO data, the number of times that the χ^2 statistic must be evaluated is found to be less significant than this threshold on average, and so this method significantly reduces the computational cost of the pipeline. However, there are still periods of time where the data quality is poor and the FFT method is more efficient. Consequently, the pipeline described here computes N_P and uses either the single-trigger or FFT method, depending on the threshold determined by Eq. 14.

Using one week of LIGO data from the S6 run, we find that the total cost of the matched filtering stage of the pipeline when the FFT method is used to compute the chi-squared test is 1430 CPU days. The new algorithm used by this pipeline

reduces the computational cost to 500 CPU days. For comparison, the method of computing the chi-squared test only for coincident triggers [15] costs 530 CPU days. The new algorithm allows the pipeline to compute the chi-squared statistic for *all* single-detector triggers and hence perform significantly more time-shifts to measure the noise background. The pipeline described here also allows the number of bins p in Eq. 4 to be a template-dependent quantity. We show in Sec. 4 that this can result in an improvement in the search sensitivity, especially for low-mass systems which spend many cycles in the detector's sensitive band.

3.5. Coincidence Test

To reduce the rate of false signals, we require that a signal is seen with consistent parameters in all of the detectors in the network. The pipeline enforces this requirement by performing a *conicidence test* on triggers produced by the matched filter. Triggers that pass this test are considered to be candidate events. Each single-detector trigger is labeled with the time of the trigger and the intrinsic parameters of the template that generated it. For a two detector network, signals must be seen in both detectors within a 15 ms window. This includes 10 ms for the maximum time it would take the wave to travel to both detectors and a 5 ms padding to account for timing errors. Since the same waveform should be observed in both detectors, the pipeline requires that the parameters of the best-fit template are consistent in the network.

Previous searches used a metric-based coincidence test [60] that checked the template parameters of triggers are sufficiently similar to each other, but did not require that they are the same. This metric-based test was developed as previous searches used different template banks in each detector to account for differences in the detector noise. Since the PyCBC pipeline uses the same template bank to matched filter the data from each detector, it implements a stricter test, called *exact-match* coincidence. In addition to the ± 15 ms window on the trigger time of arrival, the pipeline requires that the intrinsic parameters (masses and/or spins) of the triggers in the two detectors are exactly the same. Triggers that survive this test are considered conincident events. These candidates are then ranked by the quadrature sum of the reweighted SNR in each detector. For a two-detector network, this is simply

$$\hat{\rho}_c = \sqrt{\hat{\rho}_1^2 + \hat{\rho}_2^2}. \tag{15}$$

The extension of this coincidence test to more than two detectors is straightforward, however here we only consider a two-detector network.

Unlike previous searches, no clustering is performed over the template bank on the single-detector triggers prior constructing the coincidence. The new coincidence requirement decreases the chance that triggers generated by noise transients will be found in coincidence between detectors, as it is a stricter test than the metric-based

coincidence test previously used. However, there is clustering over the template bank after coincidence testing, as described in Section 3.6. The exact-match method of testing for coincidence is useful in situations where there is no simple metric to compare gravitational waveforms, as is the case with template waveforms for binaries with spinning neutron stars or black holes [61].

3.6. Candidate Event Significance

The final step of the pipeline is to measure the false-alarm rate of the search as a function of the detection statistic $\hat{\rho}_c$. This is then used to assign a statistical significance to candidate events. The false-alarm rate of the search depends on the pipeline's response to detector noise and must be measured empirically. The false-alarm rate is measured using *time shifts*. Triggers from one detector are shifted in time with respect to triggers from the second detector, and then the coincidence step is re-computed to create a background data set that does not contain coincident gravitational-wave signals. Under the assumptions that noise in the detectors is not correlated and that gravitational-wave signals are sparse in the data set, this is an effective way to estimate the background. Repeating this procedure many times on a sufficiently large interval of input data produces a large sample of the noise triggers. This allows the pipeline to compute the false-alarm rate of the search as a function of the detection statistic. Having computed the false-alarm rate of the search, each candidate event in the coincident data set is assigned a p-value that measures its significance. A candidate event's p-value p_b is the likelihood that one or more events generated from noise could have a detection-statistic value as large or larger than the candidate event itself. We describe this procedure below.

The observation time used in the search $T_{\rm obs}$ is determined by two considerations. The maximum length of the observation time is set by the requirement that the data quality of the detectors should not change significantly during the search. If the time-shifts mix data from periods of substantially different data quality (e.g. higher than or lower than average trigger rates), the noise background may not be correctly estimated. The minimum length of the observation time is set by the minimum false-alarm rate necessary to claim the detection of interesting signals. For example, it may be required that a candidate event occurs at a detection-statistic value at which the false-alarm rate is less than one in ten thousand years. The smallest false-alarm rate that the search can measure scales as $\sim \delta/T_{\rm obs}^2$, where δ is the time-shift interval. For example, in a two-detector search using time shifts of 0.2 seconds, approximately five days of coincident data are sufficient to measure false-alarm rates of 1 in 10^4 years. Given that the duty cycle of the LIGO detectors is approximately 70%, this means that the search typically operates on $\sim$ 10 calendar days of data at a time. Adjacent time can be added to measure the search backhround to lower false-alarm rates if the quality of the data is stable over these times.

Using the distribution of coincident noise events, we can calculate how many background events, n_b, are louder than a given candidate event. We can create a map for each value of the re-weighted SNR $\hat{\rho}_c$ to the number of background events $n_b(\hat{\rho}_c)$ that have a larger detection-statistic value than the candiate event. The probability that any given candidate event has n^* or fewer noise background triggers louder than it is

$$p(n_b \leq n^* | N_e = 1, N_b)_0 = \frac{1 + n^*}{N_b}, \tag{16}$$

where N_b is the total number of background events and N_e is the number of candidate events under consideration. We can think of this proceedure as comparing a candiate event to the rank-ordered list of background events from the time-shifts. A candidate event's detection-statistic value can be greater than all of noise events, less than all of the noise events, or lie in between two noise events. There are therefore $n^* + 1$ places where a given candidate event can lie in the list of background events. Thus, the probability that the candidate event has at least one background event that is louder is

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_e = 1, N_b)_0 = \frac{1 + n_b(\hat{\rho}_c^*)}{N_b}. \tag{17}$$

We wish to find the probability that that the events does not lie above this threshold, which is given by $1 - \left[1 + n_b(\hat{\rho}_c^*)\right] / N_b$. We can then find the probability that this is true for all candidate events happening by multiplying the individual probabilities for each of the N_e events together, given by

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_e, N_b)_0 = \left(1 - \frac{1 + n_b(\hat{\rho}_c^*)}{N_b}\right)^{N_e}. \tag{18}$$

However, this requires that the candiate events are independent and single noise transients can generate many triggers across the template bank. This can causes a large number of coincident triggers to be generated which are not independent. To prevent this, the pipeline implements a second stage of clustering on coincident events to select the coincident event with the highest detection-statistic value within a time window. The window size used for this clustering is an empirically tuned parameter, typically 10 s. This clustering reduces the number of candidate events and ensures that they are independent. Therefore the probability that at least one of these N_e candidate events *is* louder than the background is then

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_e, N_b)_0 = 1 - \left(1 - \frac{1 + n_b(\hat{\rho}_c^*)}{N_b}\right)^{N_e}. \tag{19}$$

Since we do not know N_b and N_e before performing the analysis, we should treat these event counts as stochastic variables. However, since there is such a large number of background events, the statistical uncertainty on N_b is negligibly small. We expect the number of background events for the amount of background time T_b analyzed to

be roughly equal to the number of foreground events for the amount of the foreground time analyzed, T:

$$\frac{N_e}{T} \approx \frac{N_b}{T_b}. \tag{20}$$

Under the assumption that the candidate events are all due to noise, we can model N_e as a Poisson distribution with a mean of $\langle N_e \rangle_0 = (T/T_b)N_b$.

Since we would like to know the significance of a given $\hat{\rho}_c^*$ without its dependence on the number of foreground events, we marginalize over N_e to rewrite the Eq. 19:

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_e, N_b) = \sum_{N_e} p(\geq 1 \text{ above } \hat{\rho}_c^* | N_e, N_b) p(N_e | N_b). \tag{21}$$

Since the pipeline assumes the null hypothesis and that foreground events are a Poisson process that can be measured using the background events, we can find the unknown probability $p(N_e|N_b)$. Letting the Poisson rate of foreground events be μ, then the probability of getting N_e events is

$$p(N_e|N_b) = \mu^{N_e} \frac{\exp(-\mu)}{N_e!}. \tag{22}$$

Putting this into Eq. 21, we get

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_b) = \sum_{N_e} \left\{ 1 - \left[1 - \frac{1 + n_b(\hat{\rho}_c^*)}{N_b} \right]^{N_e} \right\} \mu^{N_e} \frac{\exp(-\mu)}{N_e!}. \tag{23}$$

Since the sum of a Poisson distribution over all possible events, $\sum_{N_e} \mu^{N_e} \frac{\exp(-\mu)}{N_e!}$, is 1, this simplifies to

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_b) = 1 - \sum_{N_e} \left\{ \mu \left[1 - \frac{1 + n_b(\hat{\rho}_c^*)}{N_b} \right] \right\}^{N_e} \frac{\exp(-\mu)}{N_e!}. \tag{24}$$

Next, we multiply inside the summation by $\exp[-\mu(1 - \frac{1+n_b}{N_b})] \exp[\mu(1 - \frac{1+n_b}{N_b})]$, which is equivalent to multiplying by one. We can then simplify Eq. 24 to:

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_b) = 1 - \sum_{N_e} \left\{ \mu \left[1 - \frac{1 + n_b(\hat{\rho}_c^*)}{N_b} \right] \right\}^{N_e} \frac{\exp\left[-\mu(1 - (\frac{1+n_b}{N_b})) \right]}{N_e!} \exp\left[\frac{-\mu(1+n_b)}{N_b} \right] \tag{25}$$

Setting $\hat{\mu}$ equal to $\mu[1 - (1 + n_b)/N_b]$, we can see that we again have a sum of a Poisson rate over all possible values:

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_b) = 1 - \sum_{N_e} \hat{\mu}^{N_e} \frac{\exp(-\hat{\mu})}{N_e!} \exp\left[\frac{-\mu(1+n_b)}{N_b} \right]. \tag{26}$$

Thus, all but the last term in the sum total to one, resulting in

$$p(\geq 1 \text{ above } \hat{\rho}_c^* | N_b) = 1 - \exp\left[\frac{-\mu(1+n_b)}{N_b} \right]. \tag{27}$$

which can be rewritten in turn using the Poisson rate $\mu = T(N_b/T_b)$, giving

$$p(\geq 1 \text{ above } \hat{\rho}_c^*|T_b) = 1 - \exp\left[\frac{-T(1+n_b)}{T_b}\right]. \tag{28}$$

This equation thus gives the probability of a trigger having a louder background event given the amount of background time analyzed.

To account for the search background noise varying across the target signal space, candidate and background events are divided into three search classes based on template length. The significance of candidate events is measured against the background from the same class. This binning method prevents loud candidates in one class from suppressing candidates in another class. For each candidate event, we compute the p-value p_b. This is the probability of finding one or more noise background events in the observation time with a detection-statistic value above that of the candidate event, which is defined by Eq. 28 as

$$p_b = 1 - \exp\left[\frac{-T(1+n_b(\hat{\rho}_c))}{T_b}\right], \tag{29}$$

where T is the observation time of the search, T_b is the background time, and $n_b(\hat{\rho}_c)$ is the number of noise background triggers above the candidate event's re-weighted SNR $\hat{\rho}_c$ [62]. To account for having searched multiple classes, the measured significance is decreased by a trials factor equal to the number of classes [53].

The amount of time T_b in the background can vary for each time shift depending on the gaps in data and how much of the detector's data remains in coincidence after being time-shifted. However, we do not explicitly calculate the total background time by adding up the contribution from each time shift. Instead, we approximate T_b by taking the total coincident time and noting that our analysis produces the same triggers as a circular time-shift algorithm if there were no gaps between data. We can then approximate the background time very accurately by $T_b = T^2/\delta$, where δ is the time-shift interval. In the case where the gaps between consecutive portions of data are a multiple of the time-shift interval, this is not an approximation, but will in fact give the exact amount of background time. This is evident since shifting the data by a multiple of the time-shift interval would merely reassign which triggers are associated with each time shift, but would not change the total population of background triggers.

Two computational optimizations are applied when calculating the false-alarm rate of the search. The implementation of the time-shift method used does not require explicitly checking that triggers are coincident for each time shift successively. Instead, the pipeline takes the triggers from a given template and calculates the offset in their end times from a multiple of the time-shift interval. It is then possible to quickly find the pairs of triggers that are within the coincidence window. Furthermore, the number of background triggers is strongly dependent on the

detection-statistic value. For Gaussian noise, the number of triggers will fall exponentially as a function of $\hat{\rho}_c$. At low detection-statistic values, it is not necessary to store every noise event to accurately measure the search false-alarm rate as a function of detection statistic. Instead, the pipeline is given a threshold on $\hat{\rho}_c$ and a *decimation* factor d. Below this threshold the pipeline stores one noise event for every d events, storing the decimation factor so that the false-alarm rate can be correctly reconstructed. This saves disk space and makes access to the map between detection statistic and false-alarm rate faster.

4. Comparison to Initial LIGO Pipeline

In this section, we compare the PyCBC search pipeline described in this paper to the *ihope* pipeline used in the previous LIGO and Virgo searches for compact-object binaries [15]. We focus on tuning and testing the pipeline to improve the sensitivity to binary neutron star systems, although we note that use of the pipeline is not restricted to these sources; an application of this pipeline to the search for binary black holes in Advanced LIGO data is described in Refs. [1,25]. Section 4.1 describes the method that we use to measure the sensitivity of this pipeline and Section 4.2 compares this sensitivity to that of the ihope pipeline. To compare this pipeline with the ihope pipeline, we use two weeks of data from the sixth LIGO science run and a template bank designed to search for compact-object binaries with component masses between 1 and 12 $M_\odot$ and total mass $m_1 + m_2 \leq 24\ M_\odot$, as shown in Fig. 4. We demonstrate that for binary neutron stars, this pipeline can achieve a volume sensitivity improvement of up to $\sim 40\%$ over the ihope pipeline without reducing the sensitivity to higher mass systems.

4.1. Measuring Search Sensitivity

As the primary metric of search sensitivity, we measure the sensitivity of the pipeline by finding the *sensitive volume*, which is proportional to the number of detections a pipeline will make per unit time at a given false-alarm rate, $\mathcal{F}$. This is given by:

$$V(\mathcal{F}) = \int \epsilon(\mathcal{F}; r, \Omega, \mathbf{\Lambda}) \mathcal{D}(r, \Omega, \mathbf{\Lambda}) r^2 \mathrm{d}r \mathrm{d}\Omega \mathrm{d}\mathbf{\Lambda}. \tag{30}$$

Here, $\mathbf{\Lambda}$ contains the physical parameters of a signal (in this study, $\{m_1, m_2\}$), $\mathcal{D}(r, \Omega, \mathbf{\Lambda})$ is the distribution of signals in the universe, and ϵ is the efficiency of the pipeline at detecting signals at a distance r, sky location Ω, and the false-alarm rate $\mathcal{F}$.

We estimate the sensitive volume by adding to the data a large number of simulated signals (*injections*). The detector data is re-analyzed with these simulated signals added and, for each injection, the pipeline determines if a coincident trigger is present within 1 s of the time of the injection. We measure the senstivity as a function

of false-alarm rate by placing a threshold $\mathcal{F}^*$ on the false-alarm rate. We then count the number of injections that have detection statistic value $\hat{\rho}_c$ larger than the value corresponding to the threshold $\mathcal{F}^*$, as measured using the time-shift method. If a candidate event is present above the threshold at the time of a simulated signal, the pipeline records the simulated signal as found. If no trigger is present, the injection is missed and does not contribute to the sensitive volume of the search.

At some distance $r_{\rm max}$, we expect that any signal with $r > r_{\rm max}$ will be missed. Likewise, within some distance $r_{\rm min}$ we expect that nearly every signal will be found, even at an extremely small ($\lesssim 10^{-4}$/yr) false-alarm rate threshold. These bounds depend on the physical parameters of a signal. Gravitational waves from more massive systems have larger amplitudes and can thus be detected at greater distances than less massive systems. To first order, if a binary with a reference *chirp mass* $\mathcal{M}_0 = (m_1 m_2)^{3/5}/(m_1 + m_2)^{1/5}$ is detected at a distance r_0, then a binary with arbitrary chirp mass $\mathcal{M}$ will be detected with approximately the same reweighted signal-to-noise ratio at a distance [63]

$$r = r_0(\mathcal{M}/\mathcal{M}_0)^{5/6}. \tag{31}$$

Then, for each injection having chirp mass $\mathcal{M}_i$, we scale these reference distances via Eq. 31, with $\mathcal{M}_0 = 1.4 \cdot 2^{-1/5}$. If the spatial distribution of injections is uniform in volume, this can result in an inefficient injection set; the majority of the injections will be at large distances and the efficiency will not be well sampled in the region of interest. We therefore then draw the distance uniformly between the bounds $r_{{\rm min},i}$ and $r_{{\rm max},i}$, creating an injection distribution $\mathcal{D}$ that is uniform in chirp-mass scaled distance and isotropic in sky position and binary orientation.

Since the injection distribution $\mathcal{D}$ is known, the evaluation of the integral in Eq. 30 may be calculated by the Monte Carlo method with importance sampling [64]. We approximate Eq. 30 as an average over the N_I injections performed, given by

$$V(\mathcal{F}) \approx \frac{1}{N_I} \sum_{i=1}^{N} g_i(\mathcal{F}) \equiv \langle g(\mathcal{F}) \rangle \,, \tag{32}$$

where the sampling weight for injection i is given by

$$g_i(\mathcal{F}) = \frac{4\pi}{3} \left[r_{{\rm min},i}^3 + 3 r_i^2 (r_{{\rm max},i} - r_{{\rm min},i}) \hat{\epsilon}(\mathcal{F}; \mathcal{F}_i, r_i, \Omega_i, \mathbf{\Lambda}_i) \right] . \tag{33}$$

Here, $\hat{\epsilon} = 1$ if $\mathcal{F}_i \leq \mathcal{F}$ and 0 otherwise. The error in the estimate is of the intergal is given by [64]

$$\delta V = \sqrt{\frac{\langle g^2 \rangle - \langle g \rangle^2}{N}}. \tag{34}$$

To quantify the sensitive volume of the search pipeline, we select a compact-object binary population with masses that are distributed uniformly in component

mass with $1 \leq m_1, m_2 \leq 7$. We also restrict the total masses of binaries to be $\leq 14\,\mathrm{M}_\odot$. Since we use data from the sixth LIGO science run, $r_{\mathrm{min}} = 0.5\,\mathrm{Mpc}$ and $r_{\mathrm{max}} = 30\,\mathrm{Mpc}$ are reasonable bounds for a binary in which both component masses are $1.4\,\mathrm{M}_\odot$.

4.2. Relative Search Sensitivity and Computational Cost

As a baseline for comparison with the new pipeline, we compute the sensitivity of the ihope pipeline configured in the same way as was used in the S6/VSR2,3 search for low-mass binaries [15]. We allow template banks to extend to a total mass of $25\,\mathrm{M}_\odot$, as shown in Figure 4. Injections are generated at 3.5 PN order in the time domain using the TaylorT4 approximant. In the ihope pipeline, template banks were placed using a metric accurate to 1.5 post-Newtonian order [46] and the placement technique of Ref. [47]. To track changes in the detector noise, these searches generated a new power spectral density in each detector every 2048 seconds and used this updated power spectral density to re-compute the template-space metric. For both template placement and filtering, the power spectral density is computed by averaging 15 segments of length 256 s overlapped by 128 s using the median method of Ref. [14]. No gating is applied to the input data, and the clustering methods used in the matched filter are the same as those used in Refs. [14, 47]. A fixed number of $p = 16$ bins was used for the chi-squared test. Since different power spectral densities were used in each detector, there is no guarantee that the parameters of the templates are the same, so the metric-based coincidence methods were used to determine if the same signal was observed in both detectors [60].

To compute the false-alarm rate of the search, the ihope pipeline is run in two modes: the *two-stage* mode, described in Ref. [47] where the chi-squared test is only computed for triggers that survive coincidence, and a *single-stage* mode, where the chi-squared test is computed for all triggers using the original FindChirp algorithm. The two-stage mode was the original operation mode of the ihope pipeline, but the single-stage mode can measure the false-alarm rate of the search to lower values: 1 per 1000 years for the single-stage pipeline versus 1 per year for the two-stage pipeline on one week of data. However, the computational cost of the single stage pipeline without the improved methods described here is significantly higher than the two-stage pipeline, as shown in Table 1.

For this comparison, the PyCBC pipeline is configured to use a bank placement metric accurate to 3.5 post-Newtonian order [65] (the same order as the template waveforms) using the placement methods described in Ref. [49]. The bank placement metric is generated using a noise power spectral density that is averaged over both detectors' data during the observation time. The improved clustering and power spectral density methods are used in the matched filter. Data is again analyzed in 2048 s blocks, but the power spectral density is computed by averaging 127 segments

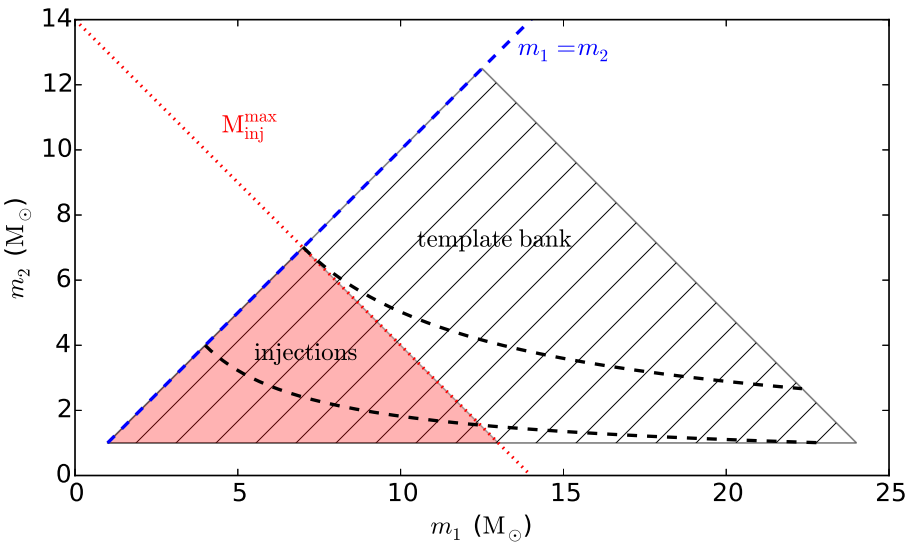

Figure 4. Mass-ranges for software injection, shown in the $m_1 - m_2$ mass-plane. The template bank used to search for these injections is indicated by hatched regions and the injection set by the red shaded region. The black dashed lines show chirp masses of 3.48 $M_\odot$ and 7.2 $M_\odot$, the boundaries between the mass bins used. Triggers from templates with chirp masses larger than 6.1 $M_\odot$ are discarded in post-processing.

Job Type	Two-Stage `ihope`	Single-Stage `ihope`	`PyCBC`
Template Bank Generation	13.3	13.3	4.7
Matched filtering and χ^2	515.4	1431	515.5
Second Template Bank	0.1	-	-
Coincidence Test	0.3	8.3	9.9
Total	529.1	1453	530.0

Table 1. The computational costs of different parts of the single-stage and two-stage ihope search pipelines, and the new PyCBC pipeline. The costs are given in CPU days.

of length 16 s, overlapped by 8 seconds. The number of chi-squared bins is allowed to vary across the search space, with $p = 100$ bins used for templates with a chirp mass $\mathcal{M} \leq 1.74 M_\odot$ and $p = 16$ bins for templates with higher chirp mass. A fixed 1 s clustering window is applied to the SNR time series to generate triggers. Using the same template bank between detectors allows us to use the exact-match coincidence test, described in Section 3.5. Two mass bins are used to estimate the noise background: a bin for triggers with $\mathcal{M} \leq 1.74 M_\odot$ and a second for triggers with $\mathcal{M} \leq 6.1 M_\odot$. Templates outside these bins are ignored in the search. A 0.2 s time-shift interval is used to measure the noise background of the search. This shows that the computational cost of the new search is comparable to that of the two-stage

search, even though the PyCBC pipeline can estimate the false-alarm rate of the search to significantly lower values than the two-stage search.

Figures 5 and 6 illustrate the key results of this comparison for two representitive weeks of LIGO S6 data. The upper plot shows the result for a week of data taken from July 2010 and the lower plot for data taken from August 2010. The first week had larger fluctuations in the overall sensitivity, but relatively few noise transients. The latter had a consistent range but more glitches. Using these two different weeks allows us to test how the pipelines work in these two different situations. The pink (dotted) line (S6 Two Stage) shows the sensitive volume of the two-stage ihope pipeline as a function of the search false-alarm rate. It can be seen that this line terminates at a false-alarm rate of 1 event per year. This is the limit of the time-shift algorithm used in the two-stage pipeline. Using the single-stage ihope pipeline, where the chi-squared statistic is computed for all triggers, this line (blue, dashed) can be extended to false-alarm rates of 1 in 1000 years. The PyCBC pipeline reaches this false-alarm rate at substantially lower computational cost, as shown in Table 1. Both figures shows that the sensitive volume of the new pipeline is equal to or better than that of the ihope pipeline for all signals in both weeks of data. Figure 5 shows that at a false-alarm rate of 1 in 100 years, the sensitivity improvement of the new pipeline is 20% for the first week of data and 40% for the second week of data. These sensitivity improvements are made without reducing the sensitivity to higer mass systems, as shown in Figure 6. This result is consistent with tuning specifically for binary neutron star systems and not binary black hole systems. A discussion for tuning to high-mass systems can be found in Ref. [25].

5. Conclusions

In this paper we have presented a new pipeline to search for gravitational waves from compact binaries using matched-filtering. The pipeline includes several improvements to the search method beyond those used in the pipeline created for the initial LIGO–Virgo science runs [13–15]. These improvements include: (i) gating to mitigate the effect of loud, non-Gaussian noise transients in the input data; (ii) the ability to use a single, fixed template bank for the entire analysis created using the harmonic mean of a large number of noise spectral density estimates; (iii) decoupling of the matched filtering and noise power spectral density estimation to improve the performance of the matched filter; (iv) a simpler, more efficient algorithm for identifying and clustering peaks in the matched-filter SNR time-series; (v) a more computationally-efficient implementation of the chi-squared veto that mitigates the need for the two-stage pipeline topology described in Ref. [15]; (vi) the ability to set the number of bins in the chi-squared veto as a function of the parameters of the template; (vii) a stricter coincidence test that requires that a trigger is found with exactly the same template in both detectors; and (viii) improvements to the time-

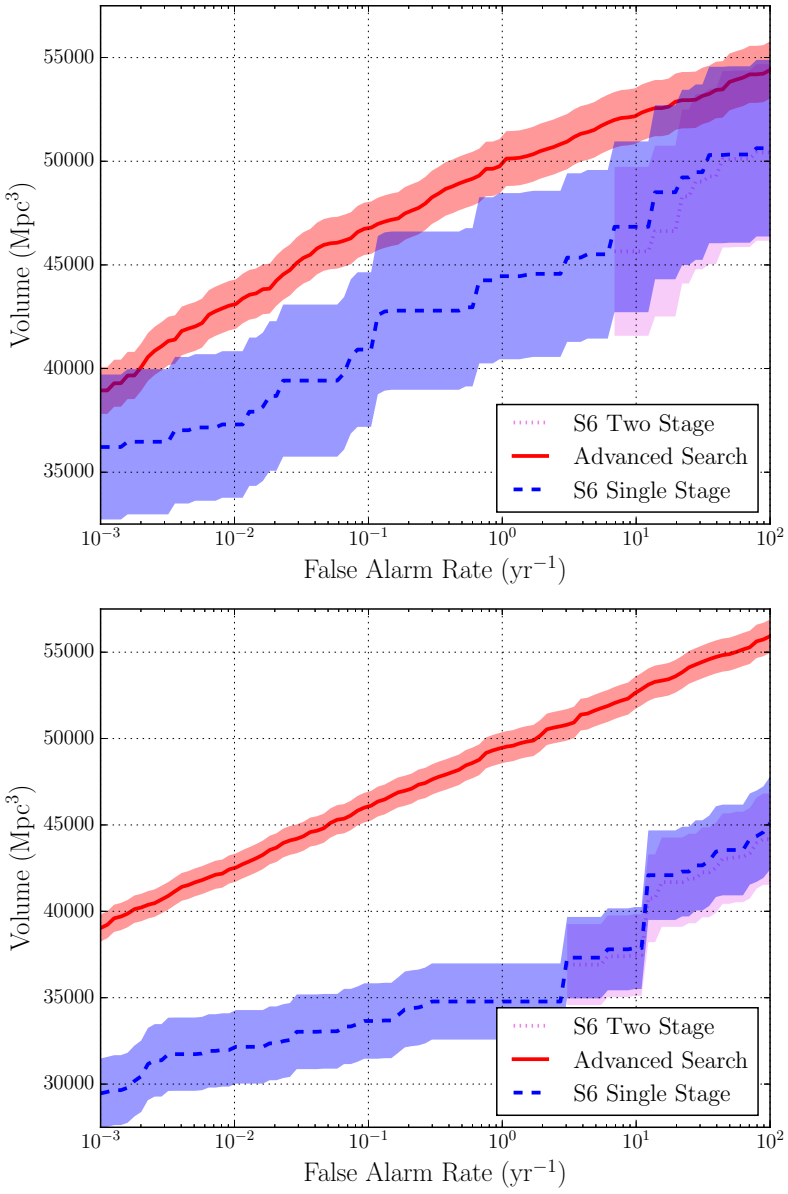

Figure 5. These plots describe the sensitivity for low-mass binary systems (chirp mass $\mathcal{M} \leq 3.48\mathrm{M}_{\odot}$). The x-axis is the significance at which injected signals can be identified, while the y-axis is the averaged volume of space the detectors could identify a simulated signal at that significance. The pink (dotted) line is the sensitivity of the two-stage pipeline used in the S6, which could not determine the significance of triggers below 6 per year. The blue (dashed) line is the analysis using a reorganized version of this pipeline, called the single-stage pipeline. The red (solid) line is the newest version of the pipeline, including a single fixed template bank that uses a harmonically averaged power spectral density and the improved χ^2 test. Notice that the old versions of the pipelines have much wider error bars since the old format was unable to perform as many injections as the new pipeline and thus cannot estimate the sensitivity as precisely. The top plot describes the first week of data from July 2010 while the bottom plot is taken from an analysis of data from August 2010.

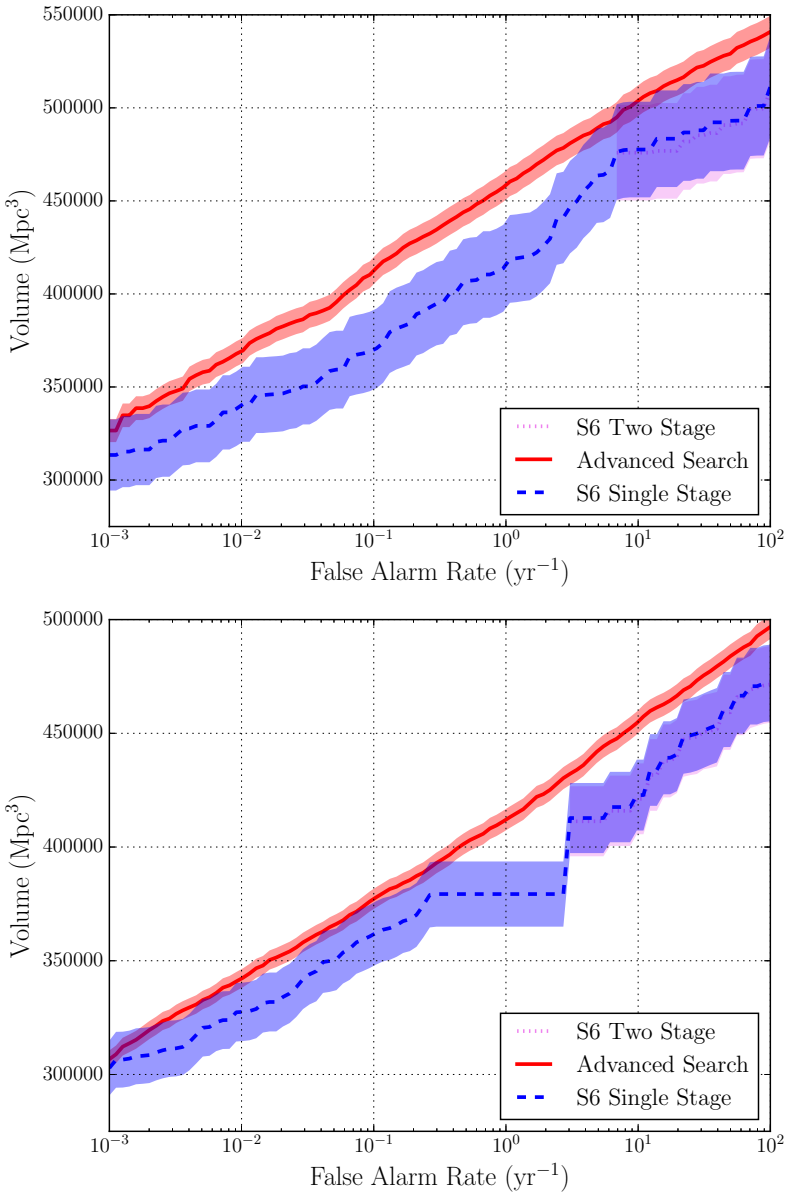

Figure 6. These plots describe the sensitivity for high-mass binary systems (chirp mass $\mathcal{M} \geq 3.48\mathrm{M}_{\odot}$). The x-axis is the significance at which injected signals can be identified, while the y-axis is the averaged volume of space the detectors could identify a simulated signal at that significance. The pink (dotted) line is the sensitivity of the two-stage pipeline used in the S6, which could not determine the significance of triggers below 6 per year. The blue (dashed) line is the analysis using a reorganized version of this pipeline, called the single-stage pipeline. The red (solid) line is the newest version of the pipeline, including a single fixed template bank that uses a harmonically averaged power spectral density and the improved χ^2 test. Notice that the old versions of the pipelines have much wider error bars since the old format was unable to perform as many injections as the new pipeline and thus cannot estimate the sensitivity as precisely. The top plot describes the first week of data from July 2010 while the bottom plot is taken from an analysis of data from August 2010.

shift method used to measure the false-alarm rate of the search and assign a p-value to candidate events. We have demonstrated that the improved methods allow up to a $\sim 40\%$ improvement in sensitivity to binary neutron star sources when tested on data from LIGO's sixth science run [52]. The pipeline described here was used in the LIGO Scientific Collaboration and the Virgo Collaboration's search for binary black holes in the first science run of Advanced LIGO [25] and to estimate the significance of the binary black hole signal GW150914 [1].

Improvements to the astrophysical sensitivity of the PyCBC search pipeline are ongoing. These developments include the extension of the pipeline to multiple detectors, upgrades to the detection statistic to include signal amplitude and phase consistency between detectors in the network and improvements to the measurement of the false-alarm rate of the search by incorporating better methods to measure how the search's false-alarm rate varies across the template bank. Ongoing advancements to the computational efficiency include investigating changes to the FFT block size and input data sample rate and the implementation of hierarchical search methods in the matched-filter. The FFT engine used in this pipeline can be efficiently used on graphics processing units (GPUs) and work is ongoing to develop a fast, efficient GPU implementation of this pipeline. Results from these studies will be reported in future publications.

References

[1] Abbott B P *et al.* (Virgo, LIGO Scientific) 2016 *Phys. Rev. Lett.* **116** 061102 (*Preprint* 1602.03837)

[2] Abbott B *et al.* (LIGO Scientific) 2009 *Rept.Prog.Phys.* **72** 076901 (*Preprint* 0711.3041)

[3] Abbott B P *et al.* (Virgo, LIGO Scientific) 2016 *Phys. Rev. Lett.* **116** 131103 (*Preprint* 1602.03838)

[4] Accadia T *et al.* (VIRGO) 2012 *JINST* **7** P03012

[5] Acernese F (Virgo) 2015 *J.Phys.Conf.Ser.* **610** 012014

[6] Akutsu T (KAGRA) 2015 *J.Phys.Conf.Ser.* **610** 012016

[7] Abbott B P *et al.* (Virgo, LIGO Scientific) 2016 (*Preprint* 1602.03842)

[8] Abadie J *et al.* (LIGO Scientific, VIRGO) 2010 *Class.Quant.Grav.* **27** 173001 (*Preprint* 1003.2480)

[9] Aasi J *et al.* (LIGO Scientific, VIRGO) 2013 (*Preprint* 1304.0670)

[10] Nitz A H, Harry I W, Willis J L, Biwer C M, Brown D A, Pekowsky L P, Dal Canton T, Williamson A R, Dent T, Capano C D, Massinger T T, Lenon A K, Nielsen A and Cabero M 2016 PyCBC Software https://github.com/ligo-cbc/pycbc

[11] Brown D A *et al.* 2004 *Class. Quant. Grav.* **21** S1625–S1633 (*Preprint* arXiv:0705.1572[gr-qc])

[12] Brown D A (LIGO) 2005 *Class. Quant. Grav.* **22** S1097–S1108 (*Preprint* gr-qc/0505102)

[13] Allen B 2005 *Phys.Rev.* **D71** 062001 (*Preprint* gr-qc/0405045)

[14] Allen B, Anderson W G, Brady P R, Brown D A and Creighton J D 2012 *Phys.Rev.* **D85** 122006 (*Preprint* gr-qc/0509116)

[15] Babak S, Biswas R, Brady P, Brown D, Cannon K *et al.* 2013 *Phys.Rev.* **D87** 024033 (*Preprint* 1208.3491)

[16] Abbott B *et al.* (LIGO Scientific) 2004 *Phys. Rev.* **D69** 122001 (*Preprint* gr-qc/0308069)
[17] Abbott B *et al.* (LIGO Scientific) 2005 *Phys. Rev.* **D72** 082001 (*Preprint* gr-qc/0505041)
[18] Abbott B *et al.* (LIGO Scientific) 2006 *Phys. Rev.* **D73** 062001 (*Preprint* gr-qc/0509129)
[19] Abbott B *et al.* (LIGO Scientific) 2008 *Phys. Rev.* **D77** 062002 (*Preprint* 0704.3368)
[20] Abbott B *et al.* (LIGO Scientific) 2008 *Phys. Rev.* **D78** 042002 (*Preprint* 0712.2050)
[21] Abbott B P *et al.* (LIGO Scientific) 2009 *Phys. Rev.* **D79** 122001 (*Preprint* 0901.0302)
[22] Abbott B P *et al.* (LIGO Scientific) 2009 *Phys. Rev.* **D80** 047101 (*Preprint* 0905.3710)
[23] Abadie J *et al.* (LIGO Scientific) 2010 *Phys. Rev.* **D82** 102001 (*Preprint* 1005.4655)
[24] Abadie J *et al.* (The LIGO Scientific) 2012 *Phys.Rev.* **D85** 082002
[25] Abbott B P *et al.* (Virgo, LIGO Scientific) 2016 (*Preprint* 1602.03839)
[26] Wainstein L A and Zubakov V D 1962 *Extraction of signals from noise* (Englewood Cliffs, NJ: Prentice-Hall)
[27] Cutler C, Apostolatos T A, Bildsten L, Finn L S, Flanagan E E *et al.* 1993 *Phys.Rev.Lett.* **70** 2984–2987 (*Preprint* astro-ph/9208005)
[28] Siemens X, Allen B, Creighton J, Hewitson M and Landry M 2004 *Class. Quant. Grav.* **21** S1723–S1736 (*Preprint* gr-qc/0405070)
[29] Abadie J *et al.* (LIGO Scientific) 2010 *Nucl.Instrum.Meth.* **A624** 223–240 (*Preprint* 1007.3973)
[30] Nuttall L *et al.* 2015 *Class. Quant. Grav.* **32** 245005 (*Preprint* 1508.07316)
[31] Aasi J *et al.* (VIRGO Collaboration) 2012 *Class. Quant. Grav.* **29** 155002 (*Preprint* 1203.5613)
[32] Aasi J *et al.* (LIGO Scientific, VIRGO) 2015 *Class.Quant.Grav.* **32** 115012 (*Preprint* 1410.7764)
[33] Buonanno A, Iyer B, Ochsner E, Pan Y and Sathyaprakash B 2009 *Phys.Rev.* **D80** 084043 (*Preprint* 0907.0700)
[34] Brown D A, Kumar P and Nitz A H 2013 *Phys.Rev.* **D87** 082004 (*Preprint* 1211.6184)
[35] Nitz A H, Lundgren A, Brown D A, Ochsner E, Keppel D *et al.* 2013 *Phys.Rev.* **D88** 124039 (*Preprint* 1307.1757)
[36] Kumar P, Barkett K, Bhagwat S, Afshari N, Brown D A *et al.* 2015 (*Preprint* 1507.00103)
[37] Blanchet L 2014 *Living Rev.Rel.* **17** 2 (*Preprint* 1310.1528)
[38] Buonanno A and Damour T 1999 *Phys.Rev.* **D59** 084006 (*Preprint* gr-qc/9811091)
[39] Pan Y, Buonanno A, Buchman L T, Chu T, Kidder L E *et al.* 2010 *Phys.Rev.* **D81** 084041 (*Preprint* 0912.3466)
[40] Damour T, Nagar A and Bernuzzi S 2013 *Phys.Rev.* **D87** 084035 (*Preprint* 1212.4357)
[41] Taracchini A, Buonanno A, Pan Y, Hinderer T, Boyle M *et al.* 2014 *Phys.Rev.* **D89** 061502 (*Preprint* 1311.2544)
[42] Damour T and Nagar A 2014 *Phys.Rev.* **D90** 044018 (*Preprint* 1406.6913)
[43] Sathyaprakash B and Dhurandhar S 1991 *Phys.Rev.* **D44** 3819–3834
[44] Dhurandhar S and Sathyaprakash B 1994 *Phys.Rev.* **D49** 1707–1722
[45] Owen B J 1996 *Phys.Rev.* **D53** 6749–6761 (*Preprint* gr-qc/9511032)
[46] Owen B J and Sathyaprakash B 1999 *Phys.Rev.* **D60** 022002 (*Preprint* gr-qc/9808076)
[47] Babak S, Balasubramanian R, Churches D, Cokelaer T and Sathyaprakash B 2006 *Class.Quant.Grav.* **23** 5477–5504 (*Preprint* gr-qc/0604037)
[48] Cokelaer T 2007 *Phys.Rev.* **D76** 102004 (*Preprint* 0706.4437)
[49] Brown D A, Harry I, Lundgren A and Nitz A H 2012 *Phys. Rev.* **D86**(8) 084017
[50] Keppel D 2013 *Phys.Rev.* **D87** 124003 (*Preprint* 1303.2005)
[51] Keppel D 2013 (*Preprint* 1307.4158)
[52] Abadie J *et al.* (LIGO Collaboration, Virgo Collaboration) 2012 *Phys.Rev.* **D85** 082002 (*Preprint* 1111.7314)
[53] Lyons L 2008 *Ann. Appl. Stat.* **2** 887–915
[54] Canton T D, Bhagwat S, Dhurandhar S and Lundgren A 2014 *Class.Quant.Grav.* **31** 015016

(*Preprint* 1304.0008)
[55] Chatterji S, Blackburn L, Martin G and Katsavounidis E 2004 *Class. Quant. Grav.* **21** S1809–S1818 (*Preprint* gr-qc/0412119)
[56] Robinet F 2015 *https://tds.ego-gw.it/ql/?c=10651* URL https://tds.ego-gw.it/ql/?c=10651
[57] Kehl Marcel S 2014 *Comparison of Fixed and Regenerated Template Banks for Compact Binary Coalescence Searches with advanced LIGO* Ph.D. thesis University of Toronto
[58] Brown D A 2004 *Searching for gravitational radiation from binary black hole MACHOs in the galactic halo* Ph.D. thesis Wisconsin U., Milwaukee (*Preprint* 0705.1514) URL https://inspirehep.net/record/673638/files/arXiv:0705.1514.pdf
[59] Allen B *et al.* 1999 *Phys. Rev. Lett.* **83** 1498 (*Preprint* gr-qc/9903108)
[60] Robinson C, Sathyaprakash B and Sengupta A S 2008 *Phys.Rev.* **D78** 062002 (*Preprint* 0804.4816)
[61] Dal Canton T *et al.* 2014 *Phys. Rev.* **D90** 082004 (*Preprint* 1405.6731)
[62] Capano C, Dent T, Hu Y M, Hendry M, Messenger C and Veitch J 2016 (*Preprint* 1601.00130)
[63] Peters P C and Mathews J 1963 *Phys. Rev.* **131** 435–439
[64] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3rd Edition: The Art of Scientific Computing* 3rd ed (New York, NY, USA: Cambridge University Press) ISBN 0521880688, 9780521880688
[65] Keppel D, Lundgren A P, Owen B J and Zhu H 2013 *Phys.Rev.* **D88** 063002 (*Preprint* 1305.5381)