Generating a Template Bank for use in Medium Latency PyGRB Searches

Michael Patel

Faculty Advisor
Dr. Ryan Fisher

Physics, Computer Science and Engineering
Christopher Newport University
April 24th, 2020

I would like to thank Dr. Fisher for all of his advice throughout my undergraduate research here at CNU.

I would also like to thank Francesco Pannarale and Ian Harry for their work developing PyCBC and their advice during this project.

Work on Medium Latency PyGRB Searches is funded by NSF Grant PHY-1912599

We are grateful for the computational resources provided by Cardiff University, funded by an STFC grant supporting UK involvement in the Operation of Advanced LIGO.

This work does not reflect the scientific opinion of the LIGO Scientific Collaboration and was not reviewed by the collaboration.

Abstract

In most modern-day searches for gravitational waves a bank of template waveforms is used to find signals buried in noisy interferometer data. In this paper, we discuss generating one of these template banks for a specific search that has time limitations. By choosing the proper generation parameters, we find that in the ideal case the bank recovers over 94% of the gravitational wave signals within its generation parameters. Finally, we show that this bank functions as expected within the searches it was designed for.

1 Introduction

In 2017, the Laser Interferometer Gravitational Wave Observatory (LIGO), in collaboration with Virgo, observed a gravitational wave (GW) created from a binary neutron star (BNS) collision. It was later shown that this GW, labeled by date as GW170817, was spatially and temporally coincident with a short gamma ray burst (GRB). To this point, the progenitors of short GRBs were largely unknown, and, thus, the discovery of a BNS collision as a potential progenitor was a major discovery for the larger Astrophysics community. Since this first discovery, much work has been done in using GRBs as triggers to search for GWs.

2 Medium Latency PyGRB Searches

Today the search for GWs coincident with GRBs is split into two categories, online and offline. Online analyses are ones that happen in pseudo real-time, and are meant to serve as a precursor to more robust offline analyses, should an interesting event be observed. Both analyses are different from traditional all-sky searches because the sky location of the event is known from the GRB observations. These triggered searches can also use a coherent analysis using multiple detectors—in conjunction with the sky location—which leads to a 25% increased sensitivity towards detection over the all-sky searches [4]. Both the online and offline analyses are done using a Python library named PyGRB, which is just a component of a larger analysis framework library: PyCBC. PyGRB is a data analysis pipeline that looks for GWs in the LIGO/Virgo data, given a set of parameters that describe the GRB event and a set of GW templates to use. Specifically of interest to this research are the Medium Latency PyGRB searches which have a latency, or time between GRB event and results, of less than 24 hours. This is considered an online analysis and is accepted to be less accurate than full offline analyses. However, these analyses must still be at least 20% more sensitive than the online all-sky search (PyCBC Live), or their advantage over this search will be lost. The goal of this research is to define limits for the aforementioned parameters that will allow

the online analyses to be sensitive enough, while still completing in 24 hours.

3 Matched Filtering

Foundational to all PyCBC searches is the concept of matched-filtering. This concept is used to find a signal buried in noise by comparing the data to templates. The following simplified mathematical derivation is adapted from [11]. We have some data, $s(t)$, in the time domain, that is a combination some noise, $n(t)$, and some signal, $h(t)$, where the noise is much louder than the signal.

$$s(t) = h(t) + n(t) \text{ for } n(t) >> h(t). \tag{1}$$

We then define a correlation between the data and a filter, F, as

$$C(\tau) = \int_{-\infty}^{\infty} s(t)F(t+\tau)dt. \tag{2}$$

Since it is computationally cheaper to work in the frequency domain we use the Fourier transform to rewrite (2) as

$$C(\tau) = \int_{-\infty}^{\infty} \tilde{s}(f)\tilde{F}^*(f)\, e^{2\pi i f\tau} df. \tag{3}$$

We are primarily interested in the signal-to-noise ratio (SNR) which is conceptually represented with the Greek letter ρ as

$$\rho = \frac{Signal}{Noise}. \tag{4}$$

To find the signal that has been matched to our filter we take the expectation value of the correlation. The filter is constant so we are left with

$$Signal = \langle C(\tau)\rangle = \int_{-\infty}^{\infty} \langle\tilde{s}(f)\rangle\tilde{F}^*(f)\, e^{2\pi i f\tau} df. \tag{5}$$

At this stage we need to make two assumptions. The first assumption is not true for most real world systems, but one that we must make in order to get anything done. We will assume that the noise is completely Gaussian with a mean of zero, that is to say the expectation value of all the noise is exactly zero. The second assumption that we will make is that if a signal was present in the data it would have a definite waveform.

$$\langle \tilde{s}(f) \rangle = \langle \tilde{h}(f) \rangle + \langle \tilde{n}(f) \rangle = \langle \tilde{h}(f) \rangle + 0 = \tilde{h}(f). \tag{6}$$

Using these assumptions simplifies (5) to

$$Signal = \int_{-\infty}^{\infty} \tilde{h}(f) \tilde{F}^*(f)\, e^{2\pi i f \tau} df. \tag{7}$$

Now that we have our signal we need to get an equation for the noise. We can represent the variance in the noise—where there is no signal—as

$$\langle N^2 \rangle = \left[\langle C^2(\tau) \rangle - \langle C(\tau) \rangle^2 \right]_{h(t)=0} = \int_{-\infty}^{\infty} \langle \tilde{n}(f) \tilde{n}^*(f') \rangle \tilde{F}(f) \tilde{F}^*(f') e^{2\pi i \tau (f - f')} df\, df'. \tag{8}$$

From here we use the expectation value of the square magnitude of the noise to define a one-sided power spectral density.

$$\langle \tilde{n}(f) \tilde{n}^*(f') \rangle = \frac{1}{2} \delta(f - f') S_n(f). \tag{9}$$

Substituting this new definition into (8), and exploiting the delta function requiring the frequencies to be the same, yields

$$N^2 = \int_{-\infty}^{\infty} \frac{1}{2} S_n(f) |\tilde{F}(f)|^2 df. \tag{10}$$

We can now write the SNR, from (7) and (10), as

$$\rho = \frac{\int_{-\infty}^{\infty} \tilde{h}(f)\tilde{F}^*(f)\, e^{2\pi i f\tau} df}{\sqrt{\int_{-\infty}^{\infty} \frac{1}{2} S_n(f)|\tilde{F}(f)|^2 df}}. \tag{11}$$

To start the simplification we define an inner product between two waveforms.

$$(a|b) = 2Re \int_{-\infty}^{\infty} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)} df = 4Re \int_0^{\infty} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)} df. \tag{12}$$

Thus, we can simplify the SNR to

$$\rho = \frac{\left(\tilde{h}(f)|\tilde{Q}(f)\right)}{\sqrt{\left(\tilde{Q}(f)|\tilde{Q}(f)\right)}}, \tag{13}$$

where $\tilde{Q}(f)$ is a template waveform, defined in terms of the filter as

$$\tilde{F}(f) = \frac{\tilde{Q}(f)}{S_n(f)} e^{-2\pi i f\tau}. \tag{14}$$

The final thing that we need to do is reverse one of the assumptions that was made in (6) and replace it with another. In reality we do not know exactly what the signal looks like, or we would not be doing this derivation, so we will revert the assumption that $\langle \tilde{s}(f) \rangle = \tilde{h}(f)$. However, in reality, we also do not know what the expectation value of the strain data is, so we make the approximation that $\langle \tilde{s}(f) \rangle = \tilde{s}(f)$, since we certainly know what the strain data is. Our final result is that the SNR in some strain data, given a template waveform, is

$$\rho = \frac{\left(\tilde{s}(f)|\tilde{Q}(f)\right)}{\sqrt{\left(\tilde{Q}(f)|\tilde{Q}(f)\right)}}. \tag{15}$$

It is easy to get lost in the math, so perhaps an example is in order. The first GW detected by LIGO was GW190514, a Binary Black Hole merger (BBH). In the top plot of Figure 1,

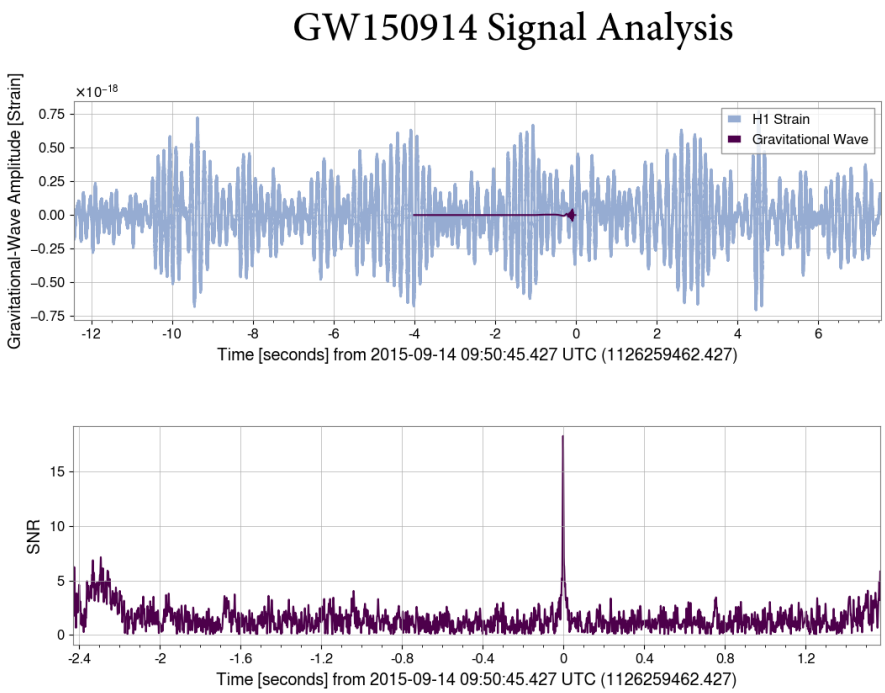

Figure 1: The top plot shows the Gravitational Wave located in the Strain data for the Hanford interferometer. The bottom plot shows the result of using matched-filtering with a single template.

the light blue is the actual strain data that we get from the Hanford interferometer. The small purple curve is the GW made by two $30M_{\odot}$ black holes (BH) colliding with each other. Without analysis algorithms, such as the matched-filter, it would be impossible to recover the GW from the noisy data. Luckily, we do have the matched filter, and, as the second plot shows, it performs admirably in detecting the GW signal. By using a template waveform and the estimation of the noise in the data, we clearly recover the GW signal with a $SNR \approx 17$.

Now that we have defined how we search for GWs and introduced the idea of templates, the impetus for this project becomes clear. We need to calculate a matched filter over many templates that are made from different parameters. This set of templates is referred to as a template bank and can contain hundreds of thousands of these templates. A large template bank allows one to cover a large parameter space, or to cover a smaller parameter space with a higher resolution. However, the catch should also be obvious, the more templates in the bank the more times one has to perform the matched filter. When designing the bank, one

must pay attention to both how well the bank recovers the signals it is searching for and how many templates are in the bank—which directly correlates to analysis computational time.

4 Generating a Noise Power Spectral Density

Before we can generate the template bank, we must get an estimation of the noise in the LIGO interferometers. This estimation is called a Noise Power Spectral Density (PSD). This is necessary because, as we discuss in Section 6, the template placement within the bank depends on the inner products defined in (12). When one computes these inner products it is often easier to use the square root of the PSD, which we call the Amplitude Spectral Density (ASD).

$$(a|b) = 4Re\int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{S_n(f)}df = 4Re\int_0^\infty \frac{\tilde{a}(f)}{\sqrt{S_n(f)}}\frac{\tilde{b}^*(f)}{\sqrt{S_n(f)}}df. \tag{16}$$

To generate the PSD, we first have to pick a time interval to run the noise estimation over. Luckily, the Gaussian noise in the LIGO detectors is fairly consistent, and so picking any decent length of data is sufficient for this estimation. For our PSD, we chose to use the data from March 8^{th} 2020, primarily because there is data for essentially the entire day. This is shown in Figure 2. Generation of the PSD is handled by a module in PyCBC. First, the module takes the data from one of the IFOs and splits it into 2048 second bins. The module then goes through each of these bins and samples the noise for each frequency every half second. For this PSD—and many others—the frequency is sampled at a rate of 256, that is to say that we estimate the noise at every $\frac{1}{256}$ Hz. Once this is done for every bin, the average is calculated for each frequency by taking the median of all of the bins. We use the median because it is less sensitive toward loud non-Gaussian glitches in one of the bins. For more information on this process, see [1]. Once we have the estimation for both detectors, the mean is taken for each frequency to generate the two-detector estimation. We then take the square root at each frequency to get the ASD. See Figure 3 for the ASD that we generated.

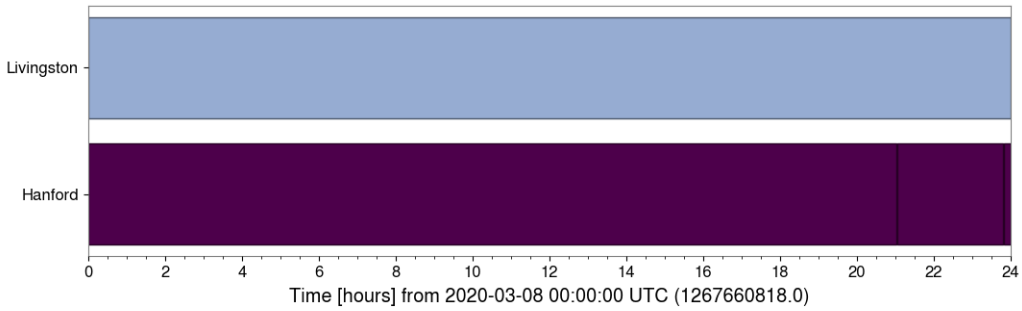

Figure 2: Observational Segments for the LIGO detectors on March 8^{th} 2020. The Livingston and Hanford interferometers were observing for 86,400 and 86,236 seconds respectively.

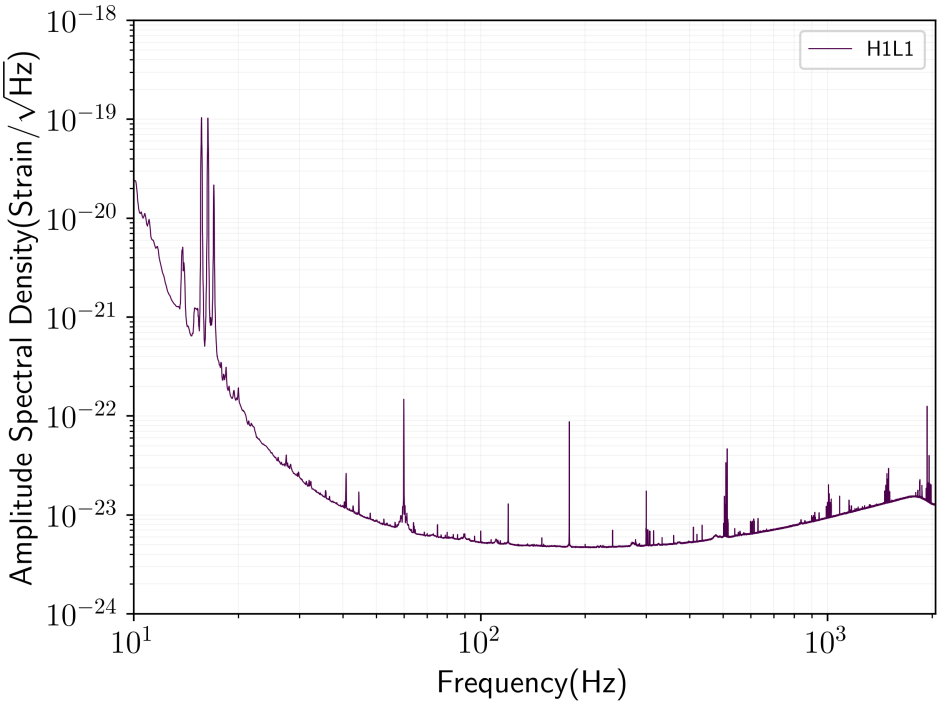

Figure 3: ASD estimation using LIGO detector data from March 8^{th} 2020. The large peaks just above 10 Hz are due to thermal noise in the suspension of the mirrors. The narrow peak at 60 Hz—and its harmonics at 120 Hz, 180 Hz, etc.—are due to power lines. For more information about the noise in the LIGO detectors, see [10]

5 Waveforms

As has been mentioned several times, we rely on the theoretical waveforms of GWs. GWs are governed by Einstein's General Theory of Relativity and are thus by no means trivial. In fact, there was no numerical solution to the two-body problem until the early $21st$ century. As such, LIGO relies on multiple methods to approximate the waveforms. Two example waveforms can be seen in Figure 4. In the top plot, the waveform is separated into three different segments, which are defined by the methods used to generate the waveform. The first section, Inspiral, is determined using Post-Newtonian approximations of the two-body problem. The second section, Merger, is calculated using numerical relativity. The final section, Ringdown, is calculated using Perturbation Theory. All three of these categories are then "stitched" together to make the final waveform that is an Inspiral-Merger-Ringdown (IMR) waveform. Each of these methods, however, can be done several ways and tuned to different parameters. These methods also differ in which domain—time or frequency—they are originally derived in. All of these differences lead to many different waveform families. There is no perfect waveform and new waveforms are always in the work.

For this project, we chose to use the IMRPhenomD waveform family for several reasons. The IMRPhenonD family is the final iteration of the IMRPhenom family—super-seeding IMRPhenomA, IMRPhenomB, and IMRPhenomC. This waveform family is also originally derived in the frequency domain which is good since we carry out the matched filtering in the frequency domain. Finally, this family of waveforms is tuned for small mass ratios with aligned spins, which are the types of signals that we intend to recover with the bank. All of this information points to the IMRPhenomD family being the best waveforms to use at the current time. For more information on IMRPhenomD waveforms, see [9] and [7]. However, as our detectors become more sensitive, and new waveform families are developed, this choice will certainly need to be revisited.

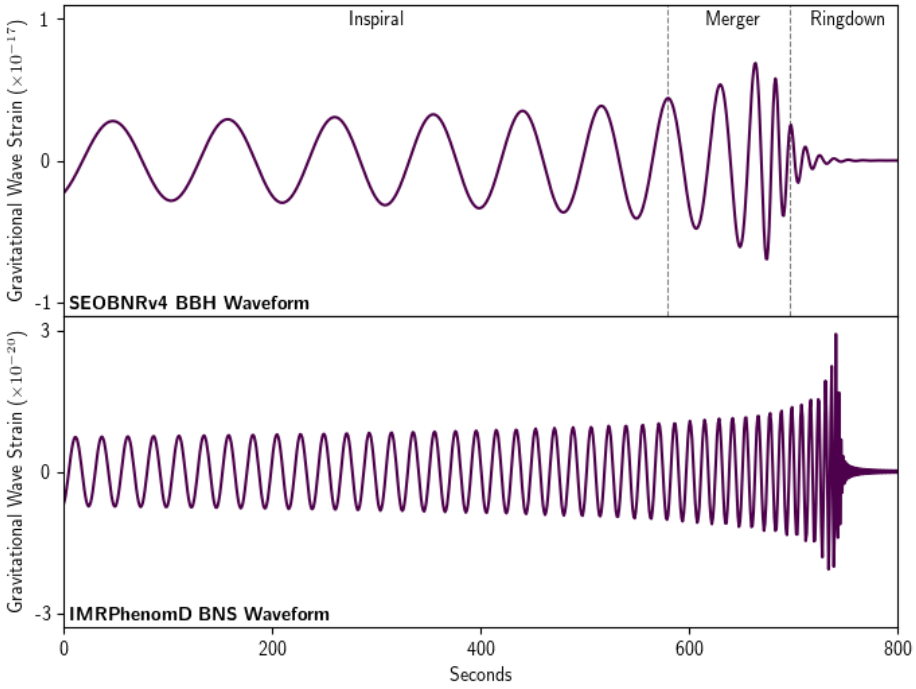

Figure 4: This figure shows two waveform approximations. The top is part of the SEOBNR_v4 waveform family, which uses an Effective One Body Approach(EOB) and is originally defined in the time domain. The bottom waveform is part of the IMRPhenomD family which is originally defined in the frequency domain.

6 Generation of a Geometric-Aligned-Spin Bank

When generating template waveforms for GWs there is one large problem. In the simplest case, a BBH, the waveforms that completely define the possible systems are part of a 15-dimensional parameter space, at minimum. These can be split into intrinsic and extrinsic parameters, and they all can be seen in Table 1. The extrinsic parameters are the parameters that define the position of the system in relation to Earth. The intrinsic parameters are the ones that describe the dynamics of the system. This 15-D parameter space is, however, only the simplest case. If we truly wanted to allow the objects to be NSs, then we must include parameters for tidal forces. The problem is that even a 15-D parameter space is, and probably will remain for a very long time, computationally infeasible. So, instead of trying

to make the system even more complicated, we instead need to start removing most of the dimensions.

Extrinsic Parameters	Intrinsic Parameters
Distance to the System	Mass_1
Right Ascension	Mass_2
Declination	Spin_{1_x}
Binary Orbital Inclination	Spin_{1_y}
Rotation between Detectors	Spin_{1_z}
Reference Time	Spin_{2_x}
Reference Phase	Spin_{2_y}
	Spin_{2_z}

Table 1: A list of the 15 parameters that fully define the simplest BBH system.

Luckily for us, most of the extrinsic parameters are taken care of by the coherent nature of our searches. We know where the GRB came from, so we have information about the sky location and how that relates to the detectors. All of these extrinsic parameters that we know get accounted for in our strain data, so the template bank does not need to focus on them. That still leaves the, at least, 8 intrinsic parameters. However, once again, we are fortunate in that waveforms are dominated by the mass terms. The effect is so dominant that, during the first observing run of LIGO, the template banks were only defined within the 2-D mass parameter space. However, as the detectors became more sensitive in later observing runs, members of LIGO discovered that template banks designed to cover the BNS range lose roughly 6% sensitivity by not accounting for spins [3]. Thus, new methods were derived to allow the bank parameter space to cover BNS systems with aligned spins—spins of both objects aligned in the positive or negative $\hat{z}$.

Currently, there are several ways of generating a template bank; the two most popular being stochastic and geometrical. The former is based on the idea of randomly placing templates in the parameter space. If one randomly adds enough templates, then teh resulting template banks will effectively cover the parameter space. The drawback of this approach is that it is random and, thus, the templates are not placed optimally within the parameter space. This leads to larger and larger banks as the required coverage is increased. In addition,

due to the random nature of the placement, these banks can take an extremely long time to generate. One such stochastic bank generated back in the second observing run required 500 CPU cores over 25 days to generate [12]. Because of these drawbacks, the approach we chose the geometric approach.

The geometric approach seeks to optimize the bank coverage and size by using a geometrical lattice in the parameter space. The chosen lattice is a flat (2-D) hexagonal-lattice. This lattice does not use the mass or spin as coordinates, but instead creates a metric based on coefficients used in the waveform approximation. The distance, or overlap, between waveforms is defined as

$$\mathcal{O}(h_1, h_2) = \frac{(h_1|h_2)}{\sqrt{(h_1|h_1)(h_2|h_2)}}, \tag{17}$$

where each of the (h_1, h_2) terms are the previously defined inner products of the waveforms. By changing the required minimum-match, or overlap, one chooses how tight the lattice must be. Once the lattice is generated the "physical coordinates" are recovered by using Monte-Carlo techniques to map points on the lattice back to the parameters such as mass and spin. If the points end up mapping to just outside the defined physical parameter space then they are "pushed in", which can be seen in Figure 5. For more information on this method, see [6].

7 Defining the Parameter Space

The next step in the generation is the definition of the parameter space that the bank needs to cover. Fundamentally, to create an electromagnetic signal the binary must contain a neutron star. So, the possible binary types to consider are neutron star-neutron star(BNS) and neutron star-black hole(NSBH). Thus, we can define parameters for the NSs and BHs based on astrophysical knowledge. The spin parameters are the easier to define, since they rely on astrophysical theory and observation. We set the unitless maximum spin magnitude —defined as $\chi = \frac{cJ}{GM^2}$ for angular momentum J—of the BH to be 0.998, known as the

"Thorne limit". For the NS, we set the unitless maximum spin to be 0.05. This value is based on NS parameters from known Galactic BNS systems and has been used many times before in LIGO analyses, see [5].

The mass parameter space is not only more important, as it is the dominant term in the waveform, but also slightly harder to pin down. We have to set the minimum mass of the NSs, the maximum masses for the BHs, and the boundary that divides them. The commonly accepted astrophysical value for minimum NS mass is just above $1M_{\odot}$. The maximum observed mass value of BHs is billions of solar masses, but most LIGO searches limit the mass to less than a hundred solar masses. The larger the BH, the faster a NSBH binary would coalesce, thus decreasing the waveform length. The boundary between the NSs and BHs is set at $2.8346480922M_{\odot}$, based on a NS equation of state originally derived in [8]. This is the boundary that has been used for all previous PyGRB searches. This boundary is the hardest one to define, because there is no definite theory on what this value should be. The value we choose is on the higher end of the proposed hypotheses and is well within the range known as the "mass gap". Current theory is unclear if these masses are NSs or BHs—if they exist at all. There are several other parameters that we use when generating the bank. These parameters all have to do with the matched-filtering inner products defined in (12). Computationally, the integral is discretized in the frequency domain. Often, instead of integrating from 0 to ∞, we choose to set definite bounds. For our searches, these values are set to be from 27Hz to 1000Hz. Both of these numbers are informed by where the detectors are most sensitive, as seen in the ASD in Figure 3, as well as by the knowledge that all BNS waveforms will coalesce before reaching 1000Hz. The last parameter that we tune is the minimum-match. This is the measure of how far one waveform is from another, and essentially defines how well the generated bank must match the parameter space.

After generating and testing 18 banks with various mass parameters, we decided that we would limit the bank to only include BNS templates. All of the parameters that we used in the final bank can be found in Table 2. This decision was made for two reasons. First,

Parameter	Value
Min-Mass	$1M_{\odot}$
Max-Mass	$2.8346480922M_{\odot}$
Max NS Spin	0.05
f_{low}	27 Hz
δf	0.01 Hz
f_{upper}	1000 Hz
Minimum Match	0.98

Table 2: This table shows the final parameters that define the extent of the template bank and how it is generated.

the only coincident detection of a GRB with a GW was from a BNS. Second, by allowing NSBH templates the bank grows extremely fast since the allowed BH spins are much greater than that of NS spins. By allowing for NSBH templates with BHs up to $10M_{\odot}$ the size of the bank can increase by over a hundred thousand templates. By limiting templates to BNS signals, we thus decrease the number of templates in the bank, which in turn reduces the time it takes the Medium Latency Searches to complete.

Our final bank has 33,488 templates. For comparison, the template bank that the offline PyGRB searches use covers the BNS and NSBH parameter spaces and contains 288,177 templates. Figure 5 shows how the templates are placed in the two parameter spaces. There are several interesting things that we should note about these placements. In both parameter spaces, there is a cluster of templates on the edge of the parameter space. This is due to points just outside the parameter space being pushed inside, as talked about in Section 6. We also see that in the mass parameter space, as the mass increases, the template density decreases. This is because as the mass increases the waveforms get shorter and start to overlap more with each other. If one were to generate a bank that ignored spins and covered BBH signals, they would see that the density of signals in the BBH range is much smaller.

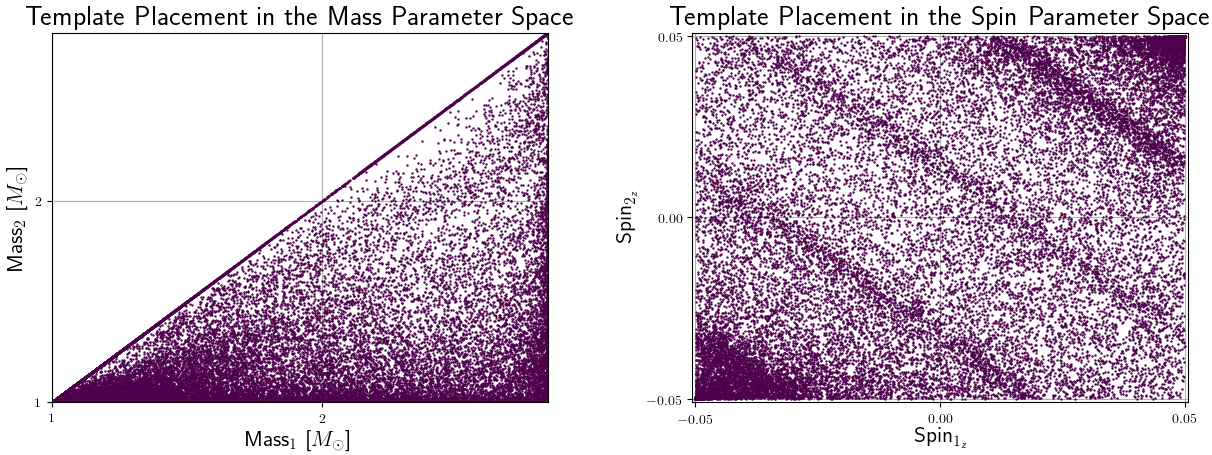

Figure 5: This figure shows how the templates are placed in the mass and spin parameter spaces. In the mass parameter space the density of templates increases around the edges of the parameter space due to templates being "pushed in". In the spin parameter space the density of templates is increased at the higher total spin values.

8 Calculating Effectiveness of the Bank

To calculate how well the bank covers the parameter space, we define a fitting-factor ($\mathcal{FF}$), first derived in [2]. This fitting-factor is simply defined as the maximum overlap (17) between the signal, h_1, and any of the template waveforms in the bank. This is represented mathematically as

$$\mathcal{FF}(h_1) = \max_{h_2 \in bank} \mathcal{O}(h_1, h_2). \tag{18}$$

To calculate fitting factors we can inject known GW signals as h_1 and determine how well the bank recovers them. We do this in two sets, broad injections and point injections.

As the name suggests, broad injections cover the entire parameter space. For the sake of this analysis, we chose to inject 10,000 signals to assess the overall performance of the bank. The result of these broad injections for our bank can be seen in Figure 6. The most important thing that the plots show is that less than 10% of the fitting factors are lower than 0.98. This value is the minimum-match value that we set at the time of bank generation, as seen in Table 1. In the bottom left plot there are two injections with the lowest fitting factors, 0.972 and 0.974 respectively. The other primary information that these plots show

is in the middle row of plots. These plots show the exact same information as the plots above, but we weight the graphing so that the lower fitting factors are plotted on top. From these plots, we can see that all the low—to the extent that 0.975 is "low"—fitting factor points are not clustered around one point in either of the parameter spaces. Thus, our bank does not have any systematic holes in its sensitivity in either of the parameter spaces. The average fitting factor for the entire bank is 0.98836, and with this value we can determine the expected sensitivity of the bank. The expected detection rate is proportional to the cube of the SNR, or fitting factor in this case [2]. Thus, we should expect this bank to recover $\sim$94% of the signals that fall inside its covered parameter space—a loss of only $\sim$6%. This expected recovery rate is not the overall sensitivity of the analysis—which depends on things such as sky-location and source distance—but an expected recovery of signals in the perfectly ideal case.

The second set of injections we do are the point injections, wherein we inject 10,000 signals at single points in the mass parameter space while varying the spins. This injection set functions primarily as a sanity check and should not show anything that we do not already know. For this set, we randomly chose 30 points in the mass parameter space. As Figure 7 shows, the lowest fitting factor of these points is 0.9880, which is consistent with the bank average from the broad injection set.

It is challenging to compare template banks against each other, as they are made with different waveforms and different search parameters. With this in mind, we attempted to see how our bank compared to PyCBC Live's bank. To do this we ran their bank through the same verification process that our bank went through. Their bank only covers the BNS region up to $2M_{\odot}$ so we had to reduce the range of injected signals for this test. According to this verification, the overall fitting factor for their bank in the BNS region was 0.9865. While technically lower, for all intents and purposes, this is the same as our average fitting factor. This similarity is encouraging because it means that our searches do not sacrifice any of their advantage over the all-sky searches in the BNS region.

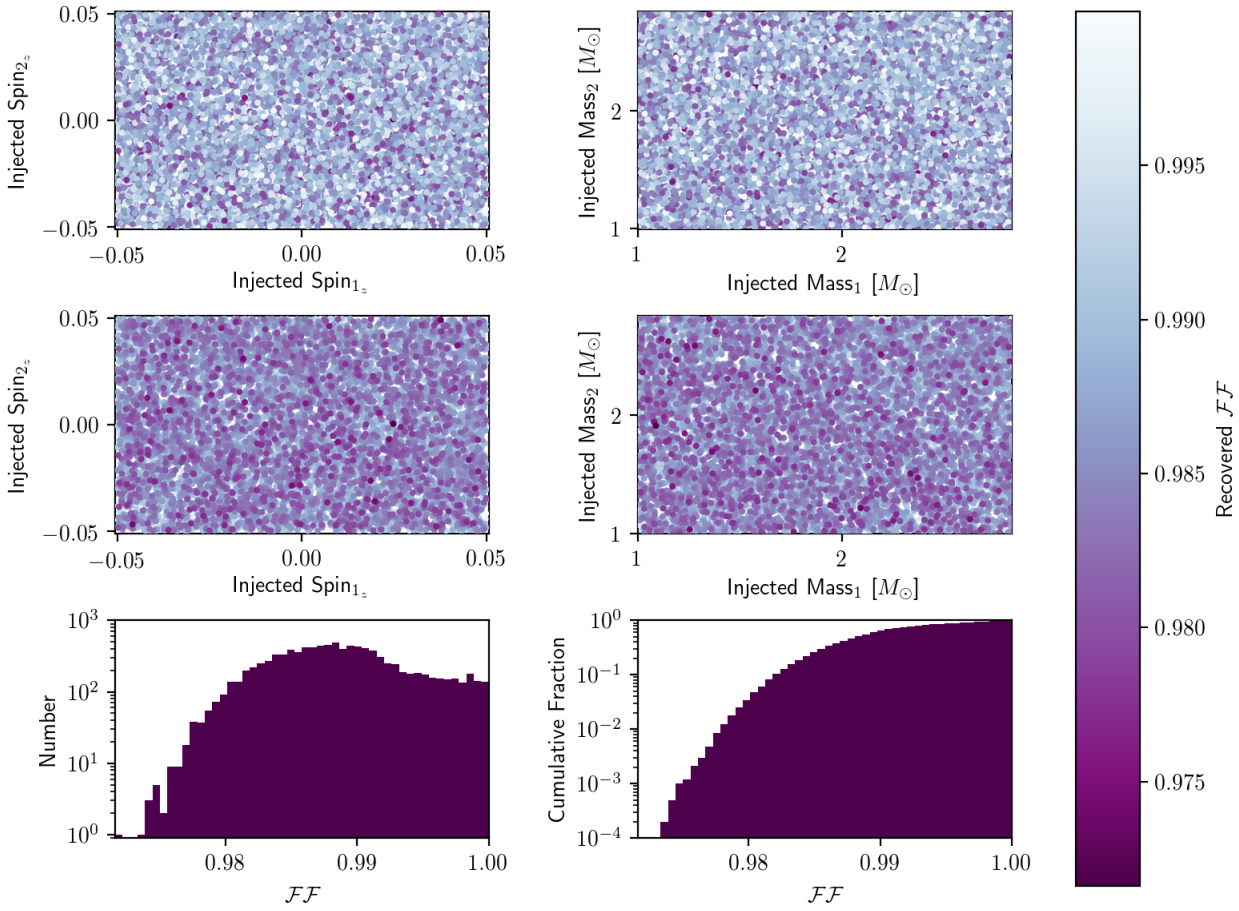

Figure 6: These plots show the results of the Broad Injection set. From top left to bottom right the plots are as follows: The first plot is of the fitting factors over the spin parameter space. The next plot is the fitting factors over the mass parameter space. The third plot contains the exact same data as the first, but it weights the fitting factors so that lower fitting factors appear on the top layer. Similarly, plot four has the same data as plot two where the fitting factors are again weighted. Plot five is a histogram of number of fitting factors per bin of fitting factors. The last plot has all the same data as the previous but is cumulative to determine percentages.

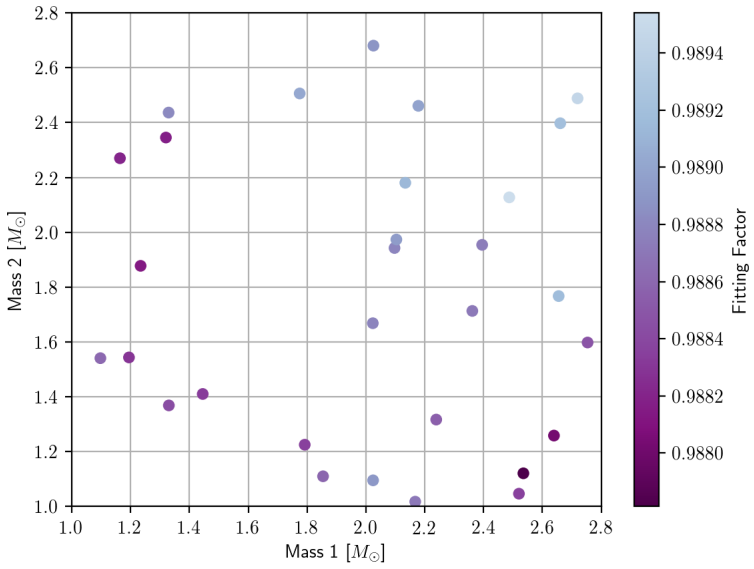

Figure 7: This plot shows the Fitting Factors for the set of point injections over the mass parameter space. At each point 10,000 signals, with their respective masses, are injected with varying spins. The color of the dot shows the average Fitting Factor for each of these points.

9 Using the Template Bank in Medium Latency

Now that we have generated our template bank, and verified that it theoretically recovers the signals we would expect, we can use it in the Medium Latency Infrastructure and see how it performs. Part of the PyGRB analysis performs the same method of injections as discussed in Section 8. The difference is that, this time, the injections are added into real detector strain data. The templates are then used to recover the injected signal within any non-Gaussian noise that may be in the detector at that given time.

Figure 8 shows the results of these injection sets over four different GRB trigger times. In all the plots, the purple crosses represent injected signals that were recovered by the template bank and are louder than all background signals. Background signals are high SNR triggers where there was no injection performed. The black crosses represent either a signal that we were not able to recover or a signal that we did recover but vetoed. A trigger is vetoed if it fails one of several tests, which are designed to differentiate between the non-

Gaussian noise and actual GW signals. One of these tests is a χ^2 test that essentially sees if there is any energy at frequencies higher than the frequency the template coalesces at. For example, if a BNS template coalesces before 1,000 Hz and there is still power at 1,200 Hz then the recovered trigger is probably not the injection but instead some loud instrumental glitch. The colored circles in the plots represent everything in between the two extremes. We define a False Alarm Probability (FAP) as the probability that the recovered signal is a false alarm. In all of the plots, there is a distinct line that separates the signals that we do recover versus the ones we do not. This line should be expected as the strain of GWs drops as $\frac{1}{r}$. Tangentially, this is why LIGO cannot see every GW. Most of them, by the time they reach Earth, are too small for our current detectors to measure. The range that the LIGO detectors can see is dependent on many things, including where the signal comes from in relation to the detectors, how many detectors observe the event, and the amount of noise in the detectors. In the current observing runs the range commonly extends out to 250 Mpc, but it is equally common to have a range of less than 100 Mpc. The important thing is that all of the plots show what we expect: that there is a clear distinction between the signals we can recover and those that we cannot recover. The second thing that we would expect to see is the effects of any non-Gaussian noise in our detectors. These effects are clearly visible in the top two plots. In the first, there is a band of missed injections at $t = -100s$ that correspond to an instrumental glitch, which is visible in the full analysis results. Similarly, in the second plot, there is a band of these missed injections at $t = 700s$. In the bottom two plots, the data is fairly clean and all the low distance missed injections are barely missed—as they have low FAPs. The last thing to note is that all of these Medium Latency Searches completed in less than 24 hours. The longest completion time was just over 17 hours and was a GRB trigger when both of the LIGO detectors, as well as the Virgo detector, were observing.

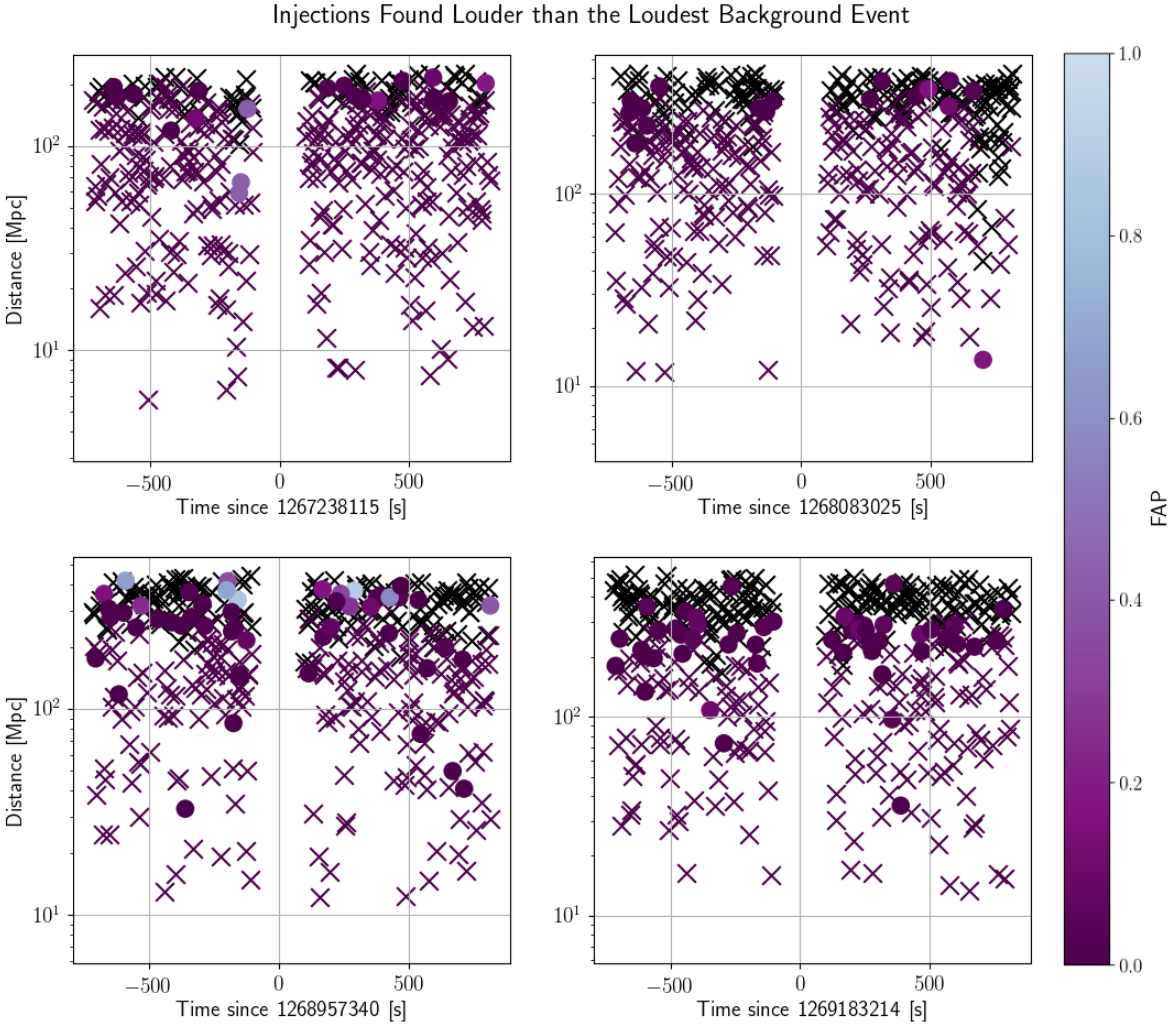

Figure 8: This figure shows the injection recovery data from Medium Latency Searches around four different GRBs.

10 Conclusion

At this point, we have achieved all the goals of this research. The bank is generated within a well defined parameter space and has been tested in the Medium Latency Searches. Furthermore, all the results suggest that the template bank is performing exactly as expected within our Medium Latency Searches. In the future, as LIGO detectors become more sensitive, there are things that may need to be adjusted. One of these things is the waveforms that we choose to use. There is now a derivative of the IMRPhenomD family, called IMRPhenomD_NRTidal, that attempts to model NS tidal forces—allowing us to treat the NSs as NSs and not miniature BHs. In future iterations of the Medium Latency Searches, it may be beneficial to use these waveforms. For now, however, the template bank functions as intended and fits the search parameters of the Medium Latency Searches.

References

[1] Bruce Allen, Warren G. Anderson, Patrick R. Brady, Duncan A. Brown, and Jolien D. E. Creighton. Findchirp: An algorithm for detection of gravitational waves from inspiraling compact binaries. *Physical Review D*, 85(12), Jun 2012.

[2] Theocharis A. Apostolatos. Search templates for gravitational waves from precessing, inspiraling binaries. *Phys. Rev. D*, 52:605–620, Jul 1995.

[3] Duncan A. Brown, Ian Harry, Andrew Lundgren, and Alexander H. Nitz. Detecting binary neutron star systems with spin in advanced gravitational-wave detectors. *Physical Review D*, 86(8), Oct 2012.

[4] J. Clark, H. Evans, S. Fairhurst, I. W. Harry, E. Macdonald, D. Macleod, P. J. Sutton, and A. R. Williamson. Prospects for joint gravitational wave and short gamma-ray burst observations. *The Astrophysical Journal*, 809(1):53, Aug 2015.

[5] Ben Farr, Christopher P. L. Berry, Will M. Farr, Carl-Johan Haster, Hannah Middleton, Kipp Cannon, Philip B. Graff, Chad Hanna, Ilya Mandel, Chris Pankow, and et al. Parameter estimation on gravitational waves from neutron-star binaries with spinning components. *The Astrophysical Journal*, 825(2):116, Jul 2016.

[6] Ian W. Harry, Alexander H. Nitz, Duncan A. Brown, Andrew P. Lundgren, Evan Ochsner, and Drew Keppel. Investigating the effect of precession on searches for neutron-star–black-hole binaries with advanced ligo. *Physical Review D*, 89(2), Jan 2014.

[7] Sascha Husa, Sebastian Khan, Mark Hannam, Michael Pürrer, Frank Ohme, Xisco Jiménez Forteza, and Alejandro Bohé. Frequency-domain gravitational waves from nonprecessing black-hole binaries. i. new numerical waveforms and anatomy of the signal. *Physical Review D*, 93(4), Feb 2016.

[8] Vassiliki Kalogera and Gordon Baym. The maximum mass of a neutron star. *The Astrophysical Journal*, 470(1):L61–L64, Oct 1996.

[9] Sebastian Khan, Sascha Husa, Mark Hannam, Frank Ohme, Michael Pürrer, Xisco Jiménez Forteza, and Alejandro Bohé. Frequency-domain gravitational waves from nonprecessing black-hole binaries. ii. a phenomenological model for the advanced detector era. *Physical Review D*, 93(4), Feb 2016.

[10] D.V. Martynov, E.D. Hall, B.P. Abbott, R. Abbott, T.D. Abbott, C. Adams, R.X. Adhikari, R.A. Anderson, S.B. Anderson, K. Arai, and et al. Sensitivity of the advanced ligo detectors at the beginning of gravitational wave astronomy. *Physical Review D*, 93(11), Jun 2016.

[11] Ed Porter. Gravitational-wave open data workshop:basics of compact binary searches, 2019.

[12] Soumen Roy, Anand S. Sengupta, and Parameswaran Ajith. Effectual template banks for upcoming compact binary searches in advanced-ligo and virgo data. *Physical Review D*, 99(2), Jan 2019.