

# Searching for gravitational waves from the coalescence of high mass black hole binaries

Second Progress Report

Ka-Lok, Lo<sup>1</sup>

Mentors: Surabhi Sachdev<sup>2</sup>, Kent Blackburn<sup>2</sup>, Alan Weinstein<sup>2</sup>

<sup>1</sup> Department of Physics, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong SAR

<sup>2</sup>LIGO Laboratory, California Institute of Technology, Pasadena, California 91125, US

July 29, 2016

Program: LIGO SURF 2016

Version 1

DCC Number: LIGO-T1600261

## Abstract

GstLAL is a search pipeline used by LIGO to search for gravitational waves from the coalescence of compact binaries (neutron stars and/or black holes), which can be operated in on-line and/or off-line configuration. Currently, the parameter space covered by GstLAL consists of binaries with component masses  $m_1, m_2 \geq 1M_\odot$  ( $M_\odot$  is the solar mass) and total mass  $2M_\odot \leq m_1 + m_2 \leq 100M_\odot$ . In this project, we aim to increase the upper bound of the total mass to  $600M_\odot$  by including intermediate mass black holes (IMBH) region into the parameter space. Simulated signal injections with real noises were used to evaluate the sensitivity of GstLAL when decreasing the lower frequency cut-off so as to observe signals from IMBH. Also, the drop in performance of the search pipeline was measured as we added more templates in the IMBH region and we experimented various techniques such as different bank splitting and  $h(t)$  gating schemes to increase the performance and sensitivity. This would enable us to detect more gravitational wave signals from the coalescence of black holes with higher masses in real time gravitational wave detection.

# 1 Objectives

The main objective of this research project is to investigate the viability of combining template bank of intermediate mass black holes (IMBH for short) with the current template bank (the uberbank) used by `gstlal`, the search pipeline that can be operated in on-line or off-line configuration.

Figure 1 shows the parameter space currently covered by `gstlal`, which consists of binary systems (Binary Black Hole, Binary Neutron Star, Neutron Star-Black Hole) with component masses  $m_1, m_2 \geq 1M_\odot$  and total mass  $2M_\odot \leq m_1 + m_2 \leq 100M_\odot$ . Intermediate mass black holes refer to black holes with masses greater than  $100M_\odot$  and the searches for IMBH are now separate and done off-line. If it is possible to include IMBH into the parameter space covered by `gstlal`, this would allow us to search for more binary black hole systems in real time.

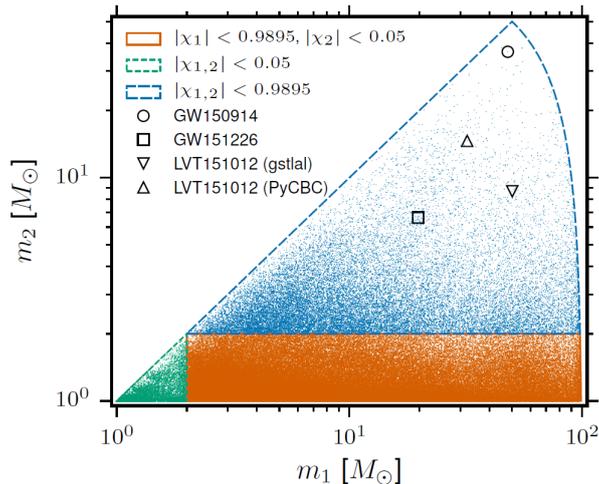


Figure 1: Parameter space currently covered by `gstlal`. (Figure taken from [2])

However, there are a lot of considerations when adding IMBH into the existing template bank of `gstlal` as this may hamper the sensitivity and performance of `gstlal`, which will be discussed in the coming sections.

## 2 $h(t)$ Gating

Under such a high sensitivity, the advanced LIGO (aLIGO) detectors are subject to tremendous amount of noises, where the noises could be fundamental, instrumental or environmental. In off-line analysis, data quality vetoes are available where poor data is flagged. Since the data  $h(t)$  is whitened and thus any two segments of  $h(t)$  are uncorrelated, segments with poor data are simply replaced by zeros [1]. However, during on-line analysis, such information is not available.  $h(t)$  gating is one of the techniques adopted by `gstlal` to eliminate short transient noises that can cause high values of signal-to-noise ratio (SNR), also known as glitches, in the data. These glitches can mimic a real gravitational wave signal and cause a false alarm trigger. Figure 2 shows an example of a glitch at time  $t = 0$  s.

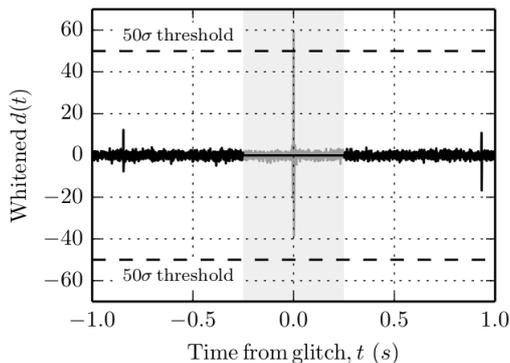


Figure 2: When the whitened  $h(t)$  exceeds a pre-defined threshold ( $50\sigma$  in this figure), the data centered at the peak with a window of 0.25 s on each side (shaded area) is regarded as a glitch and is replaced by zeros. (Figure taken from [1])

With the presence of glitches, the SNR time series would have a value higher than the expected SNR from the coalescences of binary systems that `gstlal` is aiming to detect.  $h(t)$  gating removes a segment of signals, centered at the peak of the glitch with a window of 0.25 s on each side, in a sense that it is all replaced by zeros when the SNR is greater than a threshold value as shown in Figure 2. Currently, the threshold value is usually taken as 50, which works well with the binary systems that `gstlal` is now searching for. However, since the amplitude of signal increases with the chirp mass of the binary system, real signals coming from higher mass binary systems may

have SNR higher than the threshold value that is currently used, causing `gstlal` insensitive to these higher mass binary systems. Even worse, the time a signal spent in the LIGO frequency band decreases with the chirp mass of the binary system and as a result glitches resemble to those signals and cause more false alarm triggers. An naive approach to solve this issue is to raise the threshold value such that those louder signals would not be gated out. Yet this solution is not ideal because `gstlal` would have a higher tolerance of glitchy signals in lower mass region.

Here we propose using a linear scale of  $h(t)$  gate threshold with chirp mass instead of using a single value for all masses. This simple solution addresses the issue mentioned above and now `gstlal` has different  $h(t)$  gate threshold values in different mass regions. Figure 3 shows a linear scale for  $h(t)$  gate threshold that users specify the threshold value for template with lowest chirp mass and also that for template with the highest chirp mass. The program then computes an appropriate threshold value according to this linear scale and the highest chirp mass in a sub-bank. Of course, more tests can be performed to see how we should scale the threshold value for different chirp masses to optimize sensitivity but a linear scale which is proposed here is better than using a single value for the whole region.

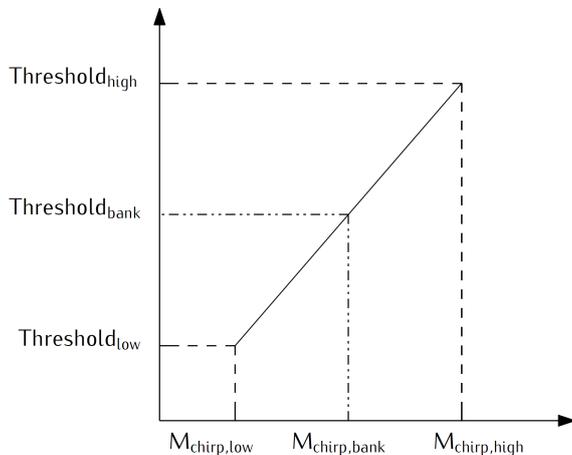


Figure 3: A linear scale for  $h(t)$  gate threshold.

This idea was implemented in the program `gstlal_inspiral` where users can either specify a single value of  $h(t)$  gate threshold value through the option `--ht-gate-threshold` or specify a linear scale of threshold value through the option `--ht-gate-threshold-linear` with the format

`mchirp_low:threshold_low-mchirp_high:threshold_high`. However, we have not been able to test this because all the cluster nodes do not have all the required plug-ins installed yet, since `gstlal` moved to using GStreamer 1.0.

### 3 Bank Splitting

Before performing singular value decomposition to the template bank, it is split into many similar sub-banks in order to optimize the compression effect and performance of singular value decomposition, which is computationally expensive. By default, the template bank is first binned by the effective spin parameter  $\chi$ . Within each bin, the templates are then sorted by the chirp mass  $\mathcal{M}$  in ascending order and  $N_T$  of the templates are grouped to form a sub-bank that should contain similar templates, as shown in Figure 4a. Finally, sub-banks are sorted in ascending order by the chirp mass of the first template. We propose several more schemes to split the template bank, which will be discussed in the coming subsections.

#### 3.1 Proposal I: Sorting by chirp time $\tau$

This proposal of bank splitter sorts templates by chirp time  $\tau$  in descending order. Chirp time  $\tau$  here refers to the time a signal staying within the LIGO frequency band. As the chirp time of a template decreases with the chirp mass of the binary system as we previously discussed, we are effectively grouping templates with similar chirp mass and chirp time together. Figure 4b shows how this bank splitter would split a template bank into many sub-banks. Intuitively, this bank splitter has the potential to outperform the default bank splitter because `gstlal_svd_bank` and `gstlal_inspiral` jobs are run in parallel and we can finish more less resource exhaustive jobs on sub-banks with higher chirp mass and shorter chirp time while the cluster is running resource exhaustive jobs on sub-banks with lower chirp mass and longer chirp time, given that not all `gstlal_inspiral` jobs are running in parallel at the same time.

In our implementation of `gstlal_bank_splitter`, the routine `lalsim_chirptime` from `SimIMRSEOBNRv2ChirpTimeSingleSpin` in `lalsimulation` library is used to compute the chirp time of a template where the routine uses thin-spline interpolation for ‘higher mass’ binary systems when the product of lower cut-off frequency and total mass in solar mass  $f_{\text{low}} \times m_{\text{total}} \geq 120$  or

3.5-PN expansion to estimate the chirp time  $\tau$  for ‘lower mass’ binary systems where the condition is not satisfied.

### **3.2 Proposal II: First binning by chirp mass then sorting by effective spin parameter $\chi$**

This proposal of bank splitter first bins templates by their chirp masses and then sorts them by their effective spin parameters  $\chi$  in ascending order. This effectively groups templates with similar  $\chi$  together. We expect that this bank splitter would perform worse than the default bank splitter as the effect of  $\chi$  to the waveform of a gravitational wave is secondary while the effect of chirp mass  $\mathcal{M}_{\text{chirp}}$  is primary.

This proposal of bank splitter was also implemented. However, there are other errors that occurred and refrained us from completing the whole `gstlal` run.

### **3.3 Proposal III: Sub-banks sorted by chirp time of the first template**

This proposal of bank splitter is basically the same as the default bank splitter except that the output sub-banks are sorted by the chirp time of their first template in descending order.

### **3.4 Performance of different bank splitters**

We can measure the compression efficiency and performance of the bank splitters in terms of latency or throughput. The following benchmarks are used to measure the latency:

1. using a program called `gstlal_inspiral_flopulator` that estimates the floating-point operations per seconds (FLOPS) needed to perform singular value decomposition to the sub-bank. Smaller the FLOPS needed, higher the compression efficiency;
2. monitoring the time required to complete `gstlal_inspiral` jobs, where matched filtering is actually done and they are the most expensive jobs in `gstlal`. Shorter the time required, higher the performance.

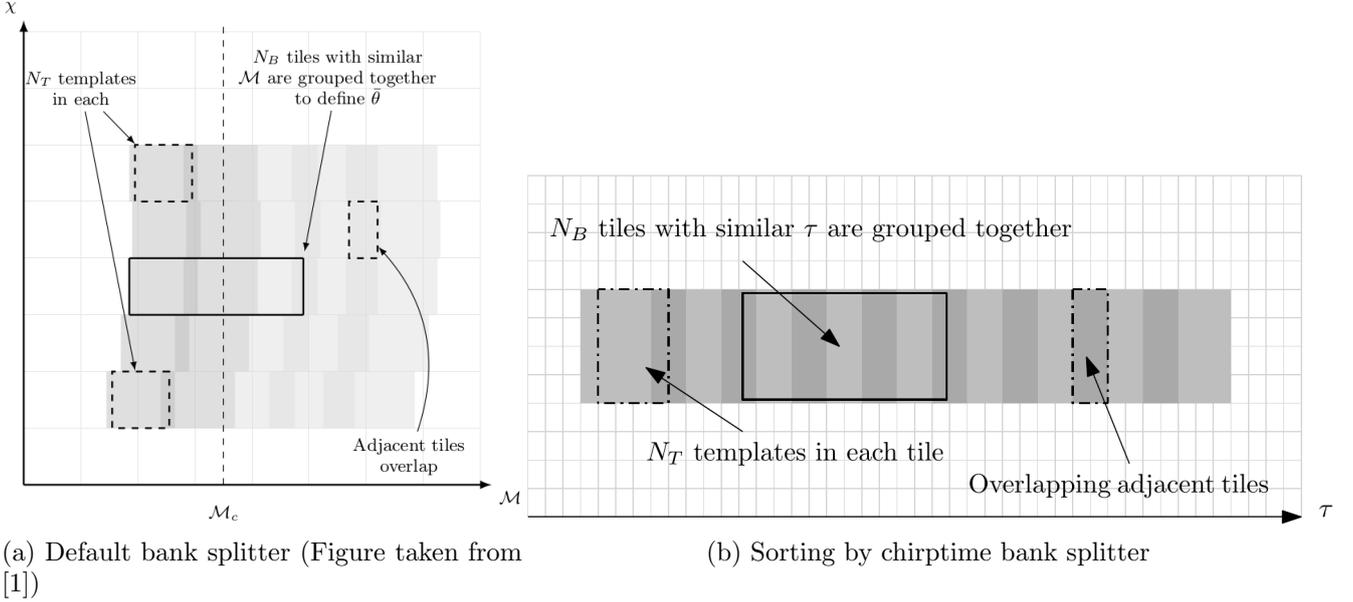


Figure 4: Different bank splitting schemes.

Figure 5a, 5b and 5c show the 2D histograms of number of floating-point operations per seconds needed to perform singular value decomposition to sub-banks split by default bank splitter, proposal I bank splitter and proposal III bank splitter respectively. We can see that the default bank splitter and proposal III bank splitter perform better than proposal I bank splitter because the total MFLOPS required by default bank splitter and proposal III bank splitter are smaller than that of proposal I bank splitter by 37.9%. Also, more sub-banks split by proposal I bank splitter require more operations sampling at the frequency of 2048 Hz.

Figure 6a and 6b show the histograms of time required to complete `gstlal_inspiral` jobs on template bank split by default and proposal I bank splitter respectively. We can see that the number of `gstlal_inspiral` jobs that take a long time ( $> 600$  minutes) for template bank split by proposal I bank splitter is smaller than that split by the default bank splitter. Also, the maximum time needed for a `gstlal_inspiral` job to complete is smaller for template bank split by proposal I bank splitter (around 600 minutes) than the default bank splitter (around 700 minutes).

## 4 Optimal SNR calculations

In the separate IMBH search, the starting frequency is lowered from 30 Hz (used by `gstlal` uberbank search) to 15 Hz. As higher the chirp mass of a binary system, the frequencies that correspond to the inspiral and merger phase decreases, this implies that if we do not lower the starting frequency when searching for IMBH, then most of the signals will be cut-off.

When  $h(t)$  only contains signals, lowering starting frequency will have a gain in SNR simply because we can now observe more parts of the signals which are previously cut off. However, this may not be true in reality because aLIGO detectors are subject to more noises at lower frequencies. Therefore, it is likely that we are adding not enough in SNR but adding a lot to our computational costs. It is also likely that we would have a loss in SNR when lowering the starting frequency. This is undesirable when we want to expand the parameter space of `gstlal` to IMBH regions as this will decrease the sensitivity of the search pipeline for binary systems with lower masses.

We can answer the question of whether the starting frequency should be lowered altogether for uberbank and IMBH region or using separate starting frequencies by doing optimal SNR calculations, which compute the SNR of various injections with a given reference power spectrum density (PSD). We investigate the optimal SNR of binary neutron star systems, binary black hole systems and neutron star-black hole systems with starting frequencies of 15, 20 and 30 Hz. The programs that do the optimal SNR calculations are still running as of today.

## 5 Modifications on `gstlal` Summary Page

In addition to reporting wall time and duty cycle (reported in the first progress report), missed/found plots in the summary page are now colored in False-Alarm Rate (FAR) and plotted in ascending order of FAR and also descending order of FAR, which are placed side by side as shown in Figure 8.

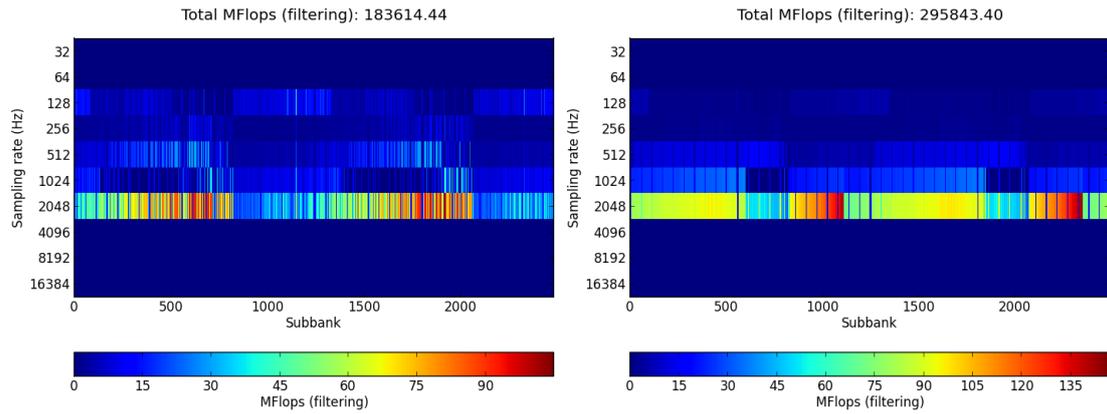
Figure 9a, 9b, 9c show an example of missed/found plots before the modification, plotted in ascending order of FAR and plotted in descending order of FAR respectively. We can see that before modification, injections are plotted in black which represents found in H1, L1 and black which represents

missed. This information cannot be conveyed by colors if injections are plotted in FAR represented by colors. The solution is using markers instead of colors: a triangle represents a missed injection whereas a circle represents found injections in H1 and L1. Moreover, with the possibility that Virgo (V1) would join our second observing run, the code is also modified such that injections found in different sets of instruments (H1,L1; H1,V1; L1,V1; H1,L1,V1) are plotted in different marker styles.

It should also be noted that there are some missed injections plotted in color while some are plotted white in color. This is because an injection is said to be missed if either the injection is found by the instruments but the FAR is higher than a threshold value specified by the user before plotting or the injection is totally missed by the instruments. The missed injections in the former case are plotted in color but with triangle marks while the latter case the injections are plotted white in color triangle marks.

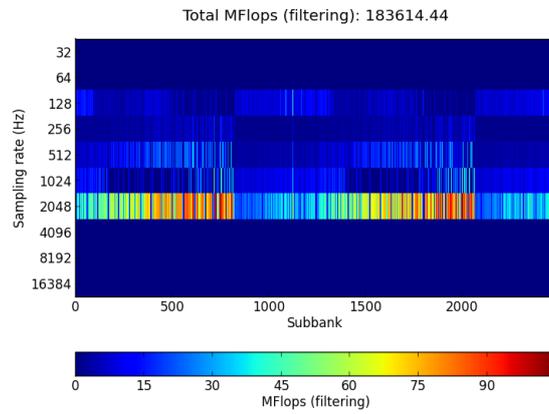
## References

- [1] C. Messick, K. Blackburn, P. Brady, P. Brockill, K. Cannon, R. Cariou, S. Caudill, S. J. Chamberlin, J. D. E. Creighton, R. Everett, C. Hanna, D. Keppel, R. N. Lang, T. G. F. Li, D. Meacher, A. Nielsen, C. Pankow, S. Privitera, H. Qi, S. Sachdev, L. Sadeghian, L. Singer, E. G. Thomas, L. Wade, M. Wade, A. Weinstein, and K. Wiesner. Analysis Framework for the Prompt Discovery of Compact Binary Mergers in Gravitational-wave Data. *ArXiv e-prints*, April 2016.
- [2] The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, and et al. Binary Black Hole Mergers in the first Advanced LIGO Observing Run. *ArXiv e-prints*, June 2016.
- [3] The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, and et al. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.*, 116:061102, Feb 2016.



(a) Default bank splitter

(b) Sorting by chirp time bank splitter



(c) Sub-banks sorted by chirp time of the first template

Figure 5: 2D histograms of MFLOPS needed to perform singular value decomposition to sub-banks split by different bank splitters as estimated by `gstlal_inspiral_floplator`.

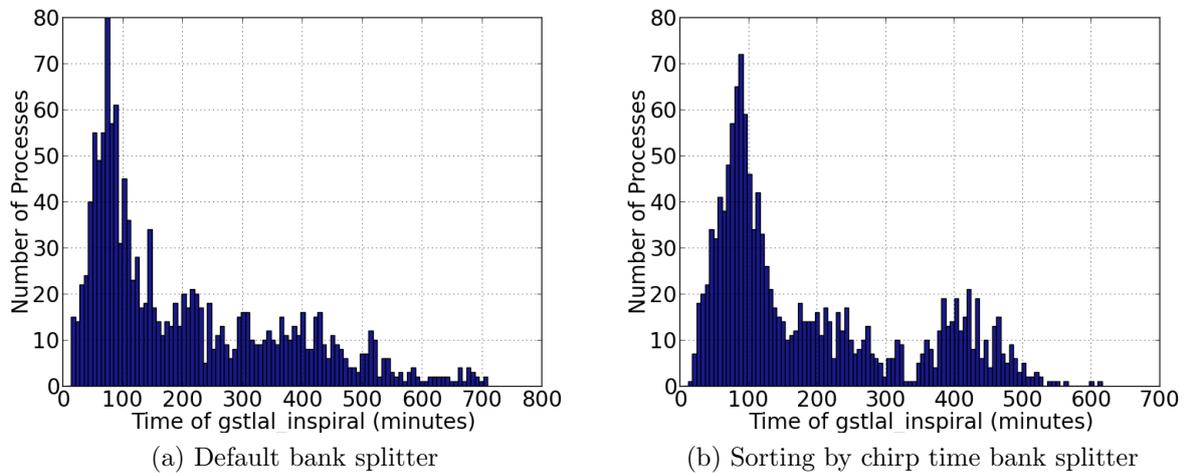


Figure 6: Histograms of time required to complete `gstlal_inspiral` jobs on template bank split by different bank splitters.

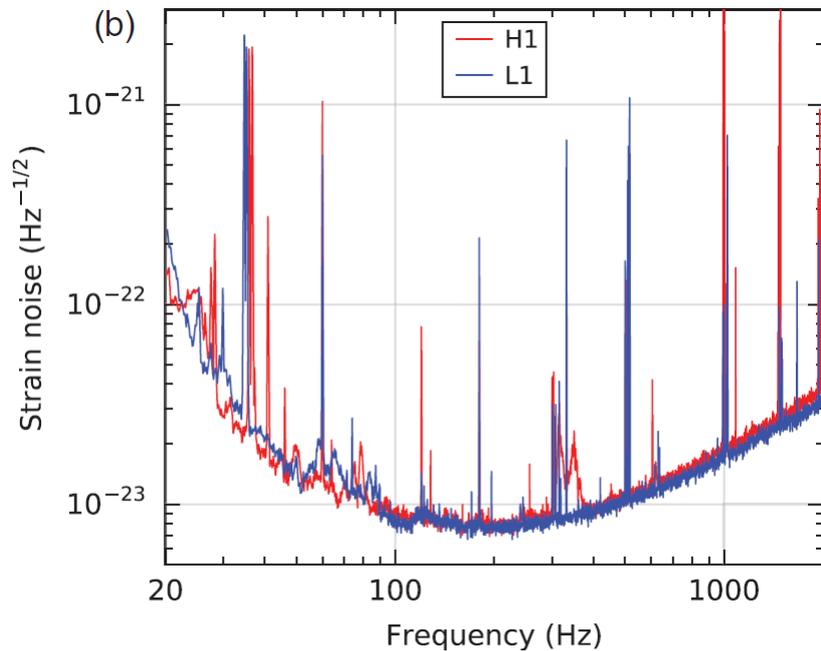


Figure 7: The amplitude spectrum density (ASD) as an indication of instrument noises of the two detectors, H1 in Hanford and L1 in Livingston, during the GW150914 event, at different frequencies. (Figure taken from [3])

## 4.1 Found / Missed Chirp Mass

Table 4.1.1. MISSED FOUND DISTANCE VS MCHIRP: ALL\_LLOID\_COMBINED

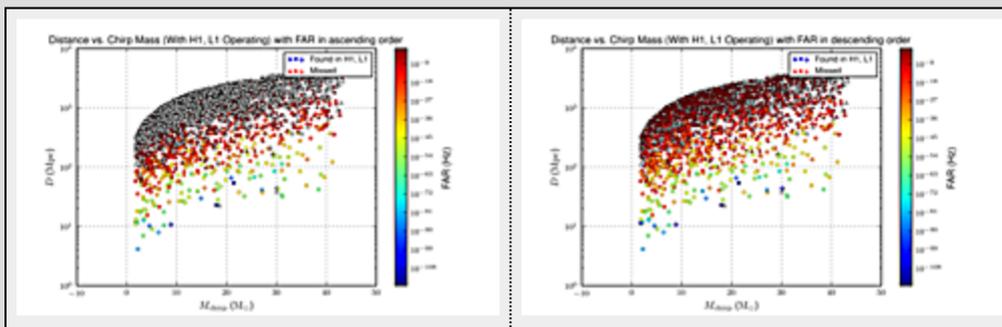
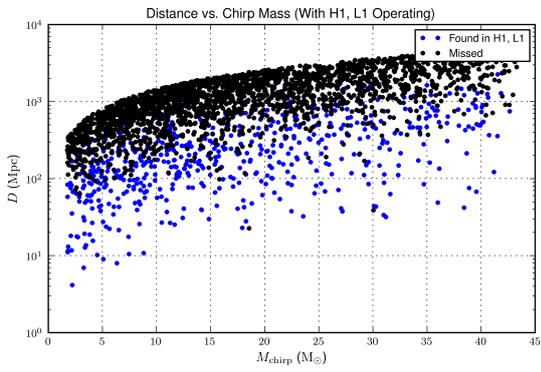
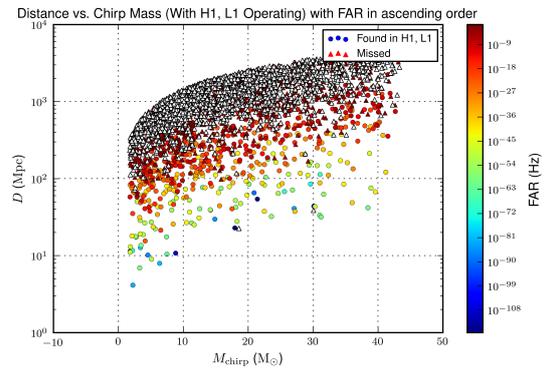


Table 4.1.1. Distance of found and missed injections as a function of chirp mass

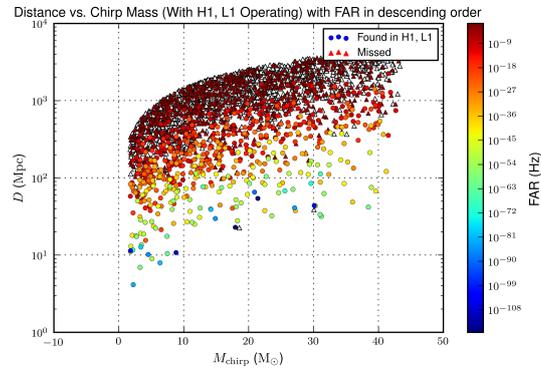
Figure 8: Missed/failed plots colored in FAR and plotted in ascending and descending order of FAR respectively.



(a) Before modification



(b) Plotted in ascending order of FAR



(c) Plotted in descending order of FAR

Figure 9: An example of missed/found plots: Distance vs. chirp mass