Investigations of Consistency in LIGO Open Science Data

Ashley Disbrow
Carnegie-Mellon University

Michele Vallisneri, Roy Williams, Jonah Kanner
California Institute of Technology

LIGO SURF Program, August 26, 2013

Abstract

The LIGO Open Science Center is preparing to release the LIGO archival data set to the public, and is developing software, tutorials, cookbooks, wikis, and teaching materials that guide the user through working with this data. We create and document a worked-out example of accessing and analyzing the LIGO Open Data Set to search for compact-binary-coalescence signals. To do so, we generate compact-binary-coalescence templates using well-modeled waveforms and perform a matched-filter search to identify LIGO hardware injections. We test for accuracy by comparing the injection signal-to-noise ratio we detect to the signal-to-noise ratio we expect to find. Of the 1492 injections that we expect to recover, 24 are not found by our search, and of the 142 injections we expect to be absent from the data, we recover 6 using our matched filter search.

1 Introduction

As requested by the NSF, the LIGO Scientific collaboration has committed to releasing its data after it's analyzed by the LSC, and after the first few confirmed detections have been found. The LIGO Open Science Center (LOSC) has been created to ensure that the large archival data sets are released to the public in a format that can be utilized in a meaningful way. Specifically, the S5 run contains the first data taken at design sensitivity from 11/2005 through 10/2007, while the S6 run contains data taken from 7/2009 to 10/2010. In preparation for the release, LOSC will archive and preserve the LIGO data by developing software, tutorials, cookbooks, wikis, and teaching materials that guide the user through working with LIGO data files, searching for binary inspirals, plotting the data, and more. One necessary part of this documentation is an accurate record of what types of hardware injections appear in the data and when.

One well-modeled potential source of gravitational waves is the Compact Binary Coalescence (CBC) of neutron stars and black holes. Throughout the S5 and S6 runs, CBC signals were mechanically injected into the data by moving the ETMs (mirrors) using magnetic actuators. These signals are useful for both the calibration of the detectors and for evaluating the efficiency of data-analysis pipelines that search for signals in the data. Users will need to be aware of the times of where these injections occur in the data so as to avoid the contamination of their searches for gravitational waves. We develop a tutorial of a CBC search, which we test by performing searches for the hardware injections. In doing so, we create an accurate list of hardware injections in the S5 data, which confirms the strength, time, location, and parameters of each injection. Section 2 will detail how we create a CBC template, perform a matched filter search, and identify whether an injection is successful or not. Section 3 contains plots of our matched filter results. Section 4 contains a discussion of how the results are interpreted, and summarizes the results from all detectors. Our conclusions are summarized in section 5. See Appendix 1 for details on working with our matched filter code, injection files, output files, etc.

2 Method for Detection of Mechanical Injections in S5 Data

The hardware injections in the S5 data comprise four different CBC models, including 1.4 – 1.4 solar mass binaries, 3 – 3 solar mass binaries, 10 – 10 solar mass binaries, and 1.4 – 10 solar mass binaries. We create template waveforms of each of these binary systems using equations 3.4 a,b, and c in Allen et. al. 2011[1]. We then use equations 4.2, 4.3, and 4.4 to match the templates with the data and normalize our results, as detailed below.

We describe injections according to their coalescence time, rather than the time of the start of the template. Thus the template starts where t<0.

Data on S5 injections comes from the documents made at the time, and is summarized at the wiki page **https://wiki.ligo.org/LOSC/S5BurstHardware** and at **https://ldas-jobs.ligo.caltech.edu/~jkanner/blue_copy/jonah_tables/cbc/**,.

We have extracted the information from H1_cbc_injections.txt, and the same for the other tw detectors. The concatenation of these lists is the basis for our study and for the list in Appendix B.

2.1 Generate CBC Templates

We generate our CBC templates in the frequency domain as detailed in equations 3.4 a-c in Allen et. al. 2011[1] for each of the different mass combinations. We set the effective distance for all of our waveforms at 1Mpc. *We found an error in equation 3.4b of Allen et. al. 2011[1] while working with our waveforms.* The exponent of the chirp mass term, $\mathcal{M}$, in equation 1 *should simplify to 5/6*, as shown in equation (2), which corrects the misprint in Ref 1.

$$\mathcal{A}_{1\,\mathrm{Mpc}}(M,\mu) = -\left(\frac{5\pi}{24}\right)^{1/2}\left(\frac{G\mathcal{M}}{c^3}\right)\left(\frac{G\mathcal{M}}{c^2 D_{\mathrm{eff}}}\right)\left(\frac{G\mathcal{M}}{c^3}\pi f\right)^{-7/6} \tag{1}$$

$$\mathcal{A}_{1\,\mathrm{Mpc}}(M,\mu) \quad \mathcal{A}_{1\,\mathrm{Mpc}}(M,\mu) = -\left(\frac{5}{24\pi}\right)^{1/2}\left(\frac{GM_\odot/c^2}{1\,\mathrm{Mpc}}\right)\left(\frac{\pi G M_\odot}{c^3}\right)^{-1/6}\left(\frac{\mathcal{M}}{M_\odot}\right)^{5/6} \tag{2}$$

When viewed in the time domain, the waveform of a CBC oscillates at a frequency in the audio band; therefore we can 'hear' a CBC signal as a chirp that whirrs from a lower frequency up to a higher frequency, as the signal becomes stronger and the amplitude of the waveform increases. In figure 1, our template waveform for a 1.4-1.4 solar mass binary shows this phenomenon occurring.

2.2 Locate Injections Using Matched Filtering

Once we have generated our templates, we use a table of injections to find the time at which each injection was made, and the parameters of the injected waveform. *We take a segment of data 64 seconds long, centered on the end time of the injection and we perform a matched filter search on the segment with the template waveform of the correct mass parameters to identify the injection's strength and location in time relative to the expected injection time.* We follow the matched filter procedure outlined in Allen et. al. 2011[1]. This is a standard data mining algorithm to find faint, but known, signals in noisy data. Equation 3 shows the complex matched filter output, z(t):

$$z(t) = 4\int_0^\infty \frac{\tilde{s}(f)\tilde{h}^*_{template}(f)}{S_n(f)} e^{2\pi i f t} df \tag{3}$$

Here, z(t) is the complex output, $\tilde{s}(f)$ is the Fourier transform of the data into the frequency domain, $\tilde{h}^*(f)$ is the complex conjugate of template in the frequency domain, and $S_n(f)$ is the power spectral density estimated from data containing no signal. Z(t) will show a sharp peak at times where the data contain a signal that is a match to the template waveform. *We find $S_n(f)$ by taking the median of the first 16 and last 16 seconds of our 64 second data segment.*

Next, the matched filter output is normalized. Our normalization constant is a match of the template with itself, considering the noise of the instrument and is shown in equation 4 below[1].

$$\sigma_m^2 = 4\int_0^\infty \frac{|\tilde{h}_{1\,\mathrm{Mpc},m}(f)|^2}{S_n(f)}df. \tag{4}$$

Here, $\tilde{h}(f)$ is the frequency domain template calculated for a CBC at a distance of 1Mpc from the detector.

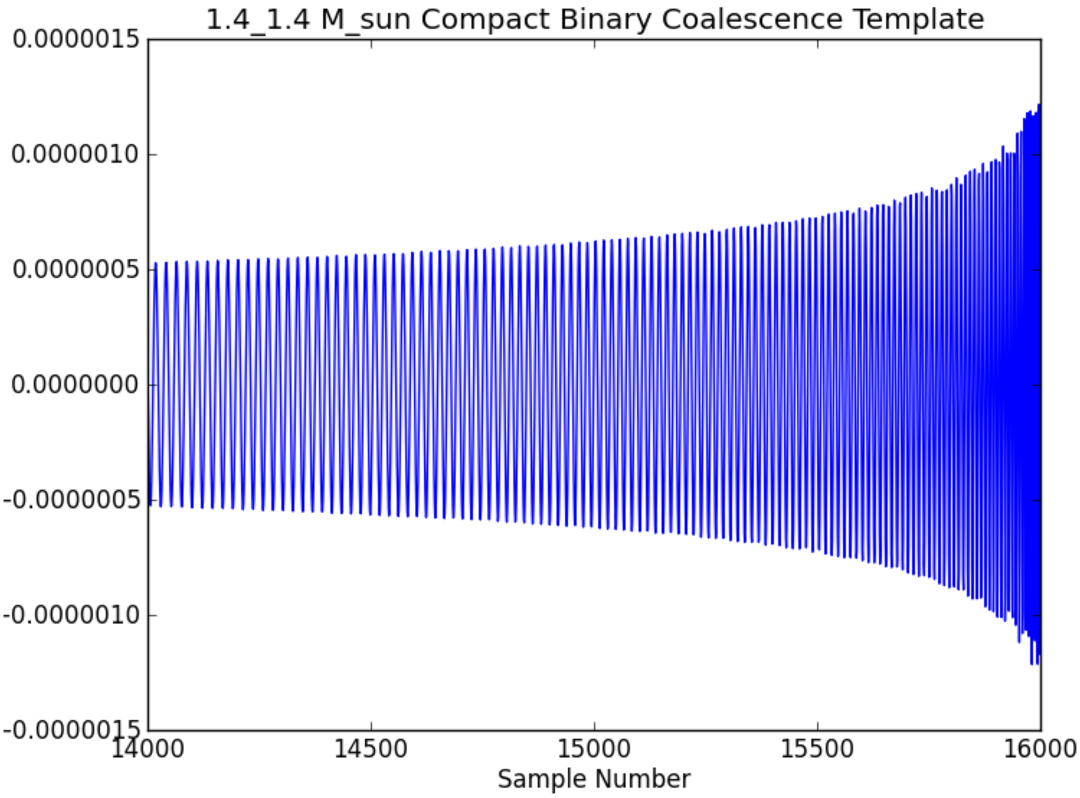

Figure 1: Template of a 1.4 – 1.4 Solar mass binary inspiral. Strain is plotted versus time, in units of 1/4096 seconds.

If we divide our matched filter output by σ_m, we obtain the amplitude signal-to-noise ratio (SNR), or recovered SNR, of the matched filter technique.

$$\rho_m(t) = \frac{|z_m(t)|}{\sigma_m} \tag{5}$$

The predicted SNR is inversely proportional to the distance to the binary system, and is shown in eqation 6.

$$\rho_E = \frac{\sigma_m}{D_{eff}} \tag{6}$$

Where ρ_E is expected SNR and D_{eff} is the effective distance to the source.

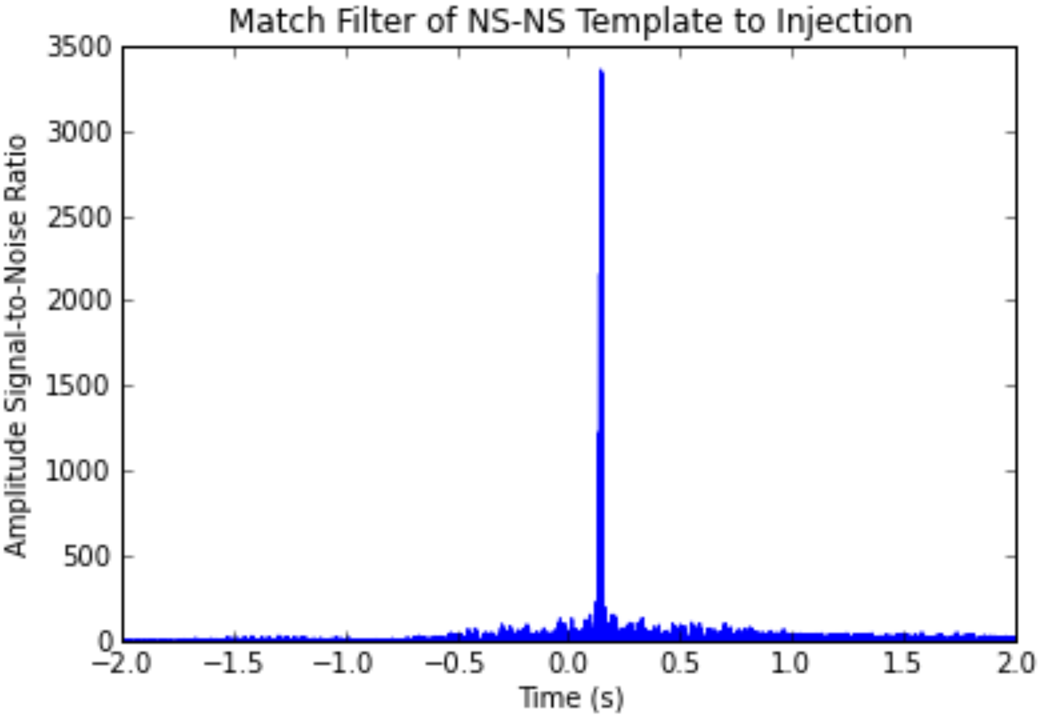

Figure 2: Amplitude SNR for 1.4 - 1.4 $M_\odot$ binary injection.

Figure 2 above shows the recovered SNR for a 1.4 – 1.4 solar mass binary as a function of time relative to the expected time of coalescence, given in the list of injections. We choose the recovered SNR to be the maximum of the recovered amplitude SNR, and the time of the detection to be the time at which the maximum SNR is found.

3 Matched Filter Results

Figures 3 – 8 show the recovered vs. predicted SNR for H1, H2 and L1. It can be seen that the 1.4 – 1.4 solar mass binaries are systematically low in all detectors. We believe that this is due to a difference in the template of the injected signal and the template we use to search for the signal. We do not include higher order corrections for the 1.4 – 1.4 binary that the hardware injections may contain.

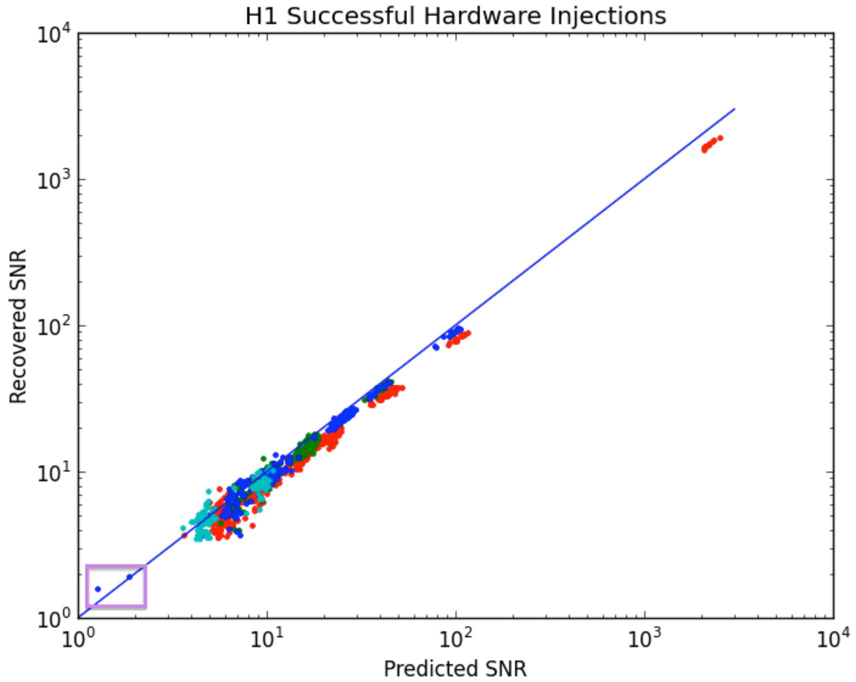

Figure 3: H1 Successful Injections Recovered vs. Predicted SNR. The purple box highlights 846225501 and 852491227

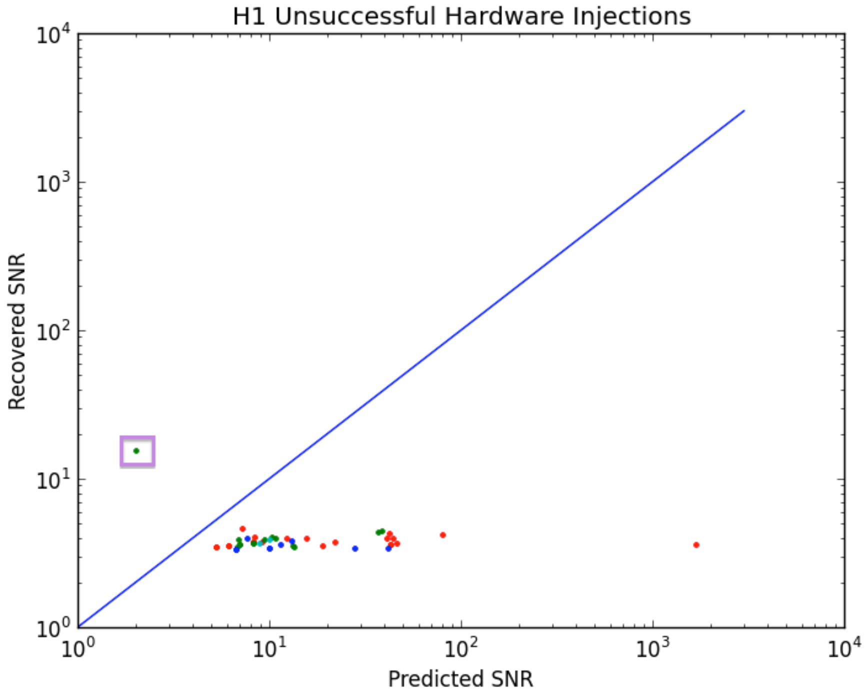

Figure 4: H1 Unsuccessful Injections Recovered vs. Predicted SNR. The purple box highlights 850065783.

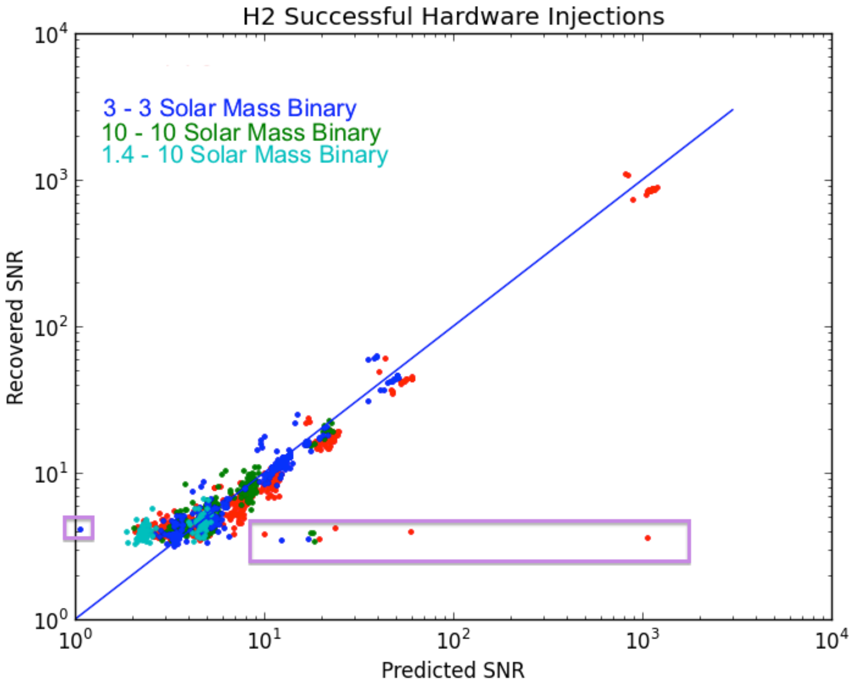

Figure 5: H2 Successful Injections Recovered vs. Predicted SNR.
The left purple box highlights ten points: 867676433.
The right box highlights 834405864, 834268673, 834484333, 834278134, 834729980, 834106623, 834448703, 834788036, 864850719, 86487205.

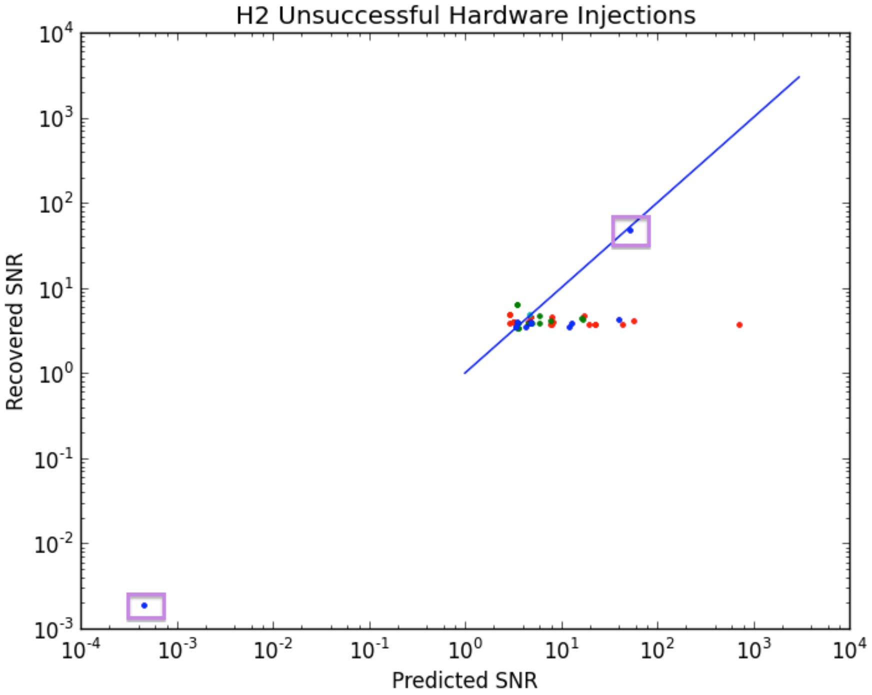

Figure 6: H2 Unsuccessful Injections Recovered vs. Predicted SNR.
The left purple box highlights 851970237.
The upper right box highlights 824214406.

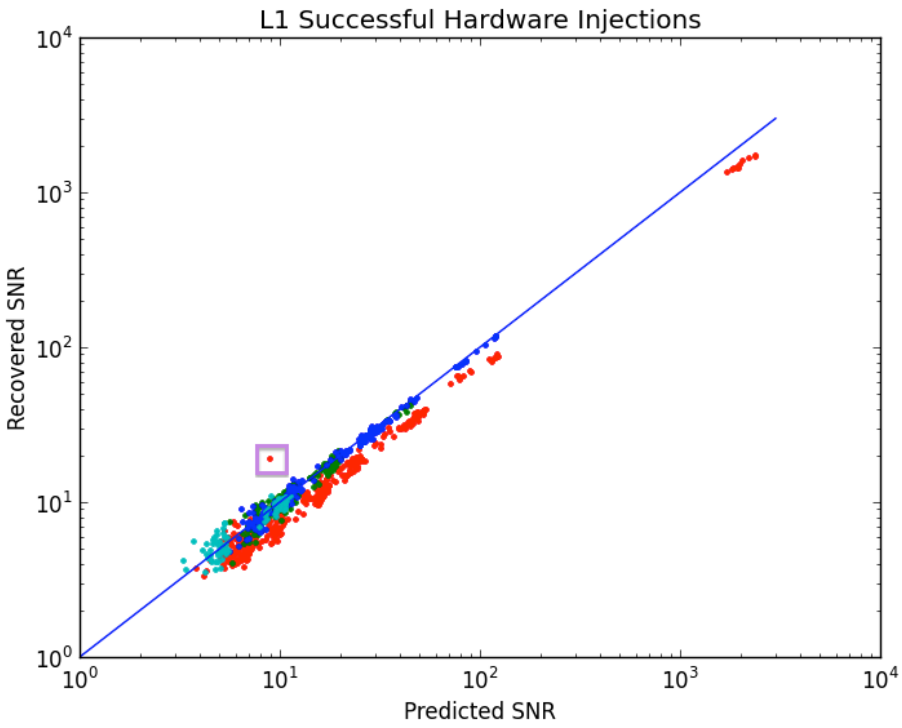

Figure 7: L1 Successful Injections Recovered vs. Predicted SNR.
The purple box highlights 869893785.

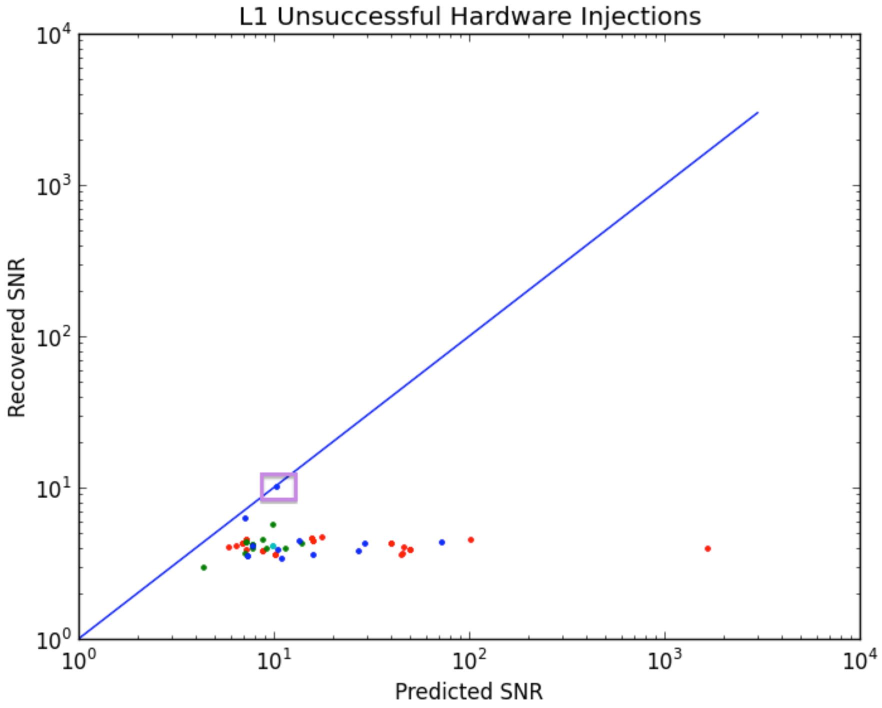

Figure 8: L1 Unsuccessful Injections Recovered vs. Predicted SNR.
The purple box highlights 852491227.

4 Discussion and Analysis

To determine the minimum SNR at which we will consider a matched filter result to be a detection, we run our matched filter code on background data for 800-900 times in each of the detectors. We histogram the recovered SNR for these background times to determine the loudest SNR recovered from noise. The histograms from each detector's background are given in figures 9 – 10 below.

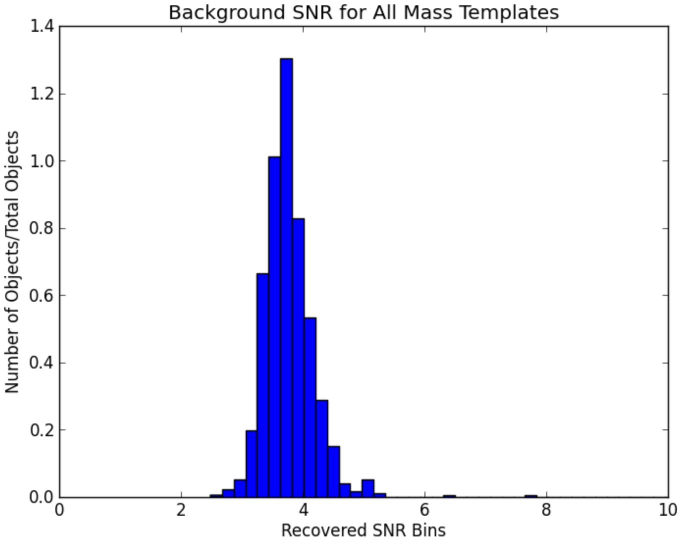

Figure 9: Recovered SNR bins for H1 background, binsize = 0.2

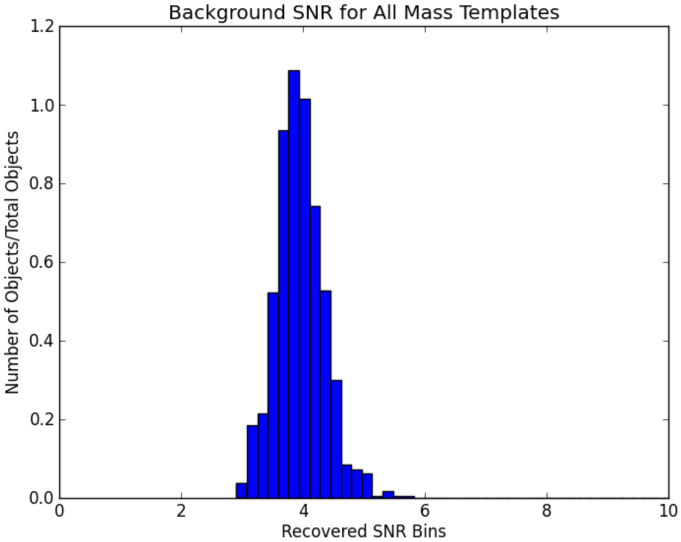

Figure 10: Recovered SNR bins for H2 background, binsize = 0.2

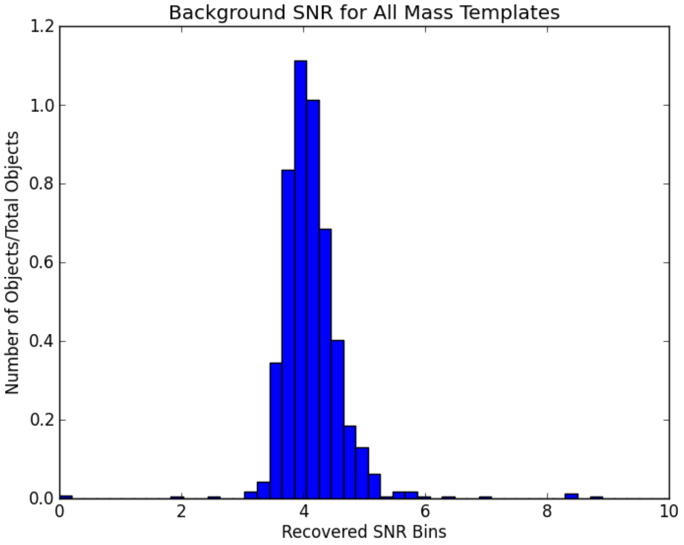

Figure 11: Recovered SNR bins for L1 background, binsize = 0.2

We choose the minimum recovered SNR to be considered as a detection for all detectors to be 6.0, since we see that most of the recovered SNR drops off after around 5.5. We find that, of the injections we expect to be loud enough to find, there are 24 we do not find, and of the unsuccessfully labeled injections, we find 6 injections we do not expect to see, as summarized in Table 1 below.

Detector	**H1**	**H2**	**L1**
Total # Injections	1200	1282	1271
Successful Injections	870 Are all of these in plots 3,5, and 7?	929	770
Successful Injections, Predicted SNR > 8	614	333	545
For Injections with Predicted SNR > 8, Injections with Recovered SNR > 6	608 Why is this less than the row above? Can you tabulate them? Do they all have predicted SNR close to 8?	322	538
Successful Injections, Data Unavailable	21 This can happen; eg, the data may be CAT1–vetoed. But	19	14

	we should check them all.		
Unsuccessful Injections, data available	46	45	51
Unsuccessful Injections with Recovered SNR > 6 Is this a subset of the row above?	1	3	2
Unsuccessful Injections, Data Unavailable	263	289	436

Table 1: Summary of classifications of predicted and recovered injections

5 Outliers

The injection at time 824214406 in the H2 detector is clearly detected using our matched filter algorithm, as it has a predicted and recovered SNR of ~50 (see Fig. 6). This injection is marked "Injection Compromised", but we successfully detect it because the detector went out of lock 38 seconds after the injection. Figure 12 shows a spectrogram of the power at each time and frequency bin of the time around the injection, which is located at time 0. This injection is a successful 10-10 solar mass binary at 10 Mpc from Earth. Could you list all the other outliers?
Especially that guy on the lower left of fig 6.

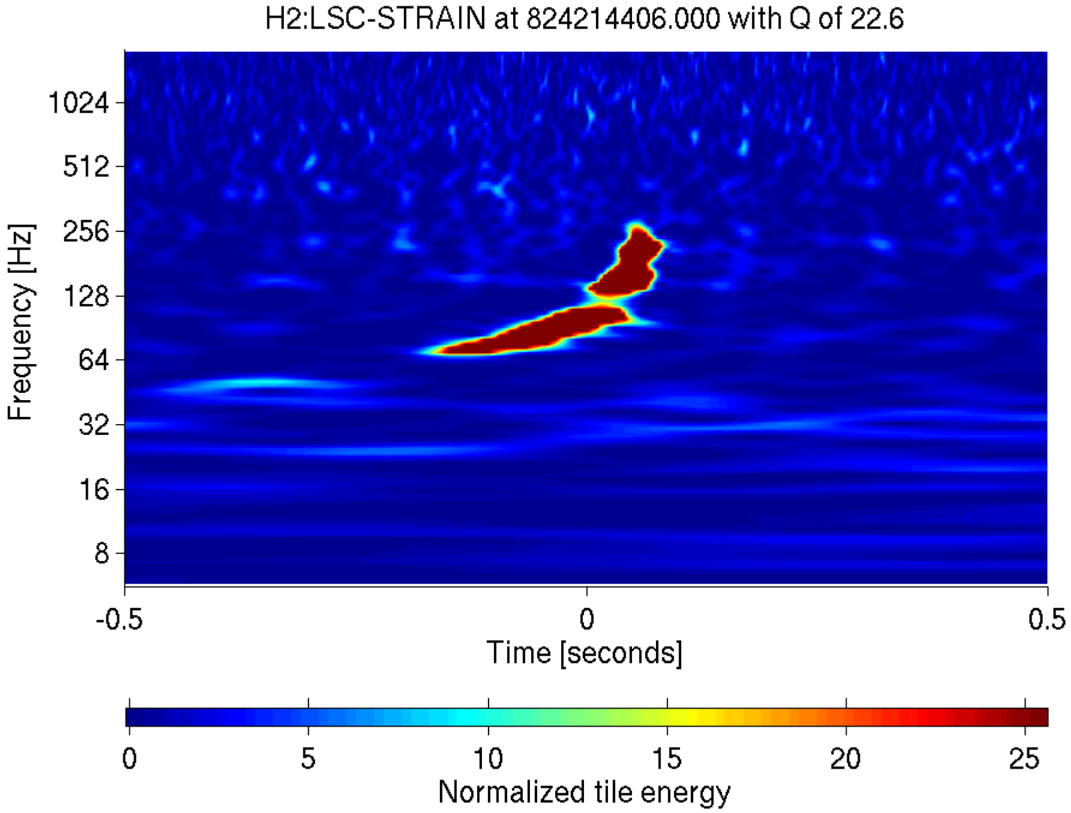

Figure 12: injection at time = 824214406 of a 10 – 10 solar mass binary at 10 Mpc. This was initially marked injection compromised, but can be detected in the data.

6 Conclusions

Most of the successful injections we expect to find are detected by our matched filter code. We find that some of them, however, are not detected as expected. We also find that some of the injections marked unsuccessful can be detected using this matched filter code. We take these results and update the injection documentation to reflect these new discoveries. With this updated documentation, LOSC users will have accurate descriptions of what they should expect to find when performing a matched filter search on hardware injections.

To-do list –

– follow up on all outliers.

– follow up on successful injections with predicted snr > 8 and recovered SNR < 6.

– follow up on successful injections with data unavailable.

– follow up on Unsuccessful Injections with Recovered SNR > 6

References

1 Bruce Allen et. al., *FINDCHIRP: an algorithm for detection of gravitational waves from inspiraling compact binaries*

Appendix A: How to use the Matched Filter Code

These instructions detail how to run the matched filter code when your current directory is checkKnownInjections/, in the SVN.

Running the matched filter code

- runMatchedFilter.py

$ python runMatchedFilter.py injection_list.csv name_of_savefile.csv detector(H1)

  - runAllInjections.sh has commands to run this file on many different input files
  - The input file is Jonah's injection file, but in CSV form. The columns are labeled left to right as: begin_inj, end_inj, m1, m2, distance, status. The file smallerInjectionFiles.py is used to create these .csv input files. ***(See the section on input files to understand what it does and how the injection statuses are encoded as numbers)***
  - The output file contains the following information: end_inj, distance, m1, m2, PredictedSNR, RecoveredSNR, time(recoveredSNR)-end_inj, status. These are the results of the matched filter output
- backgroundMatchedFilter.py
  - The same as runMatchedFilter.py, except it adds 128 seconds to the injection time.
- findchirpMatchFilter.py
  - called within runMatchedFilter.py
  - takes in some information from the injectioin file as well as a segment of the data from that hour and the quantities you define, such as the sampling frequency, and the number of seconds of data you want to search over
  - outputs the predicted and recovered SNR, rho(t), and delta t
- readligo.py
  - Code written by Jonah to read ligo HDF5 files
- templateGenFdomain.py
  - Takes the sampling frequency, length of data segment in seconds, and the two masses in the binary system and returns the template in the frequency domain and frequency

Break down injection files and plot data

- smallerInjectionFiles.py
  - when running on losc-data, $ python smallerInjectionFiles.py H1_cbc_injections.csv
  - takes Jonah's injection files and breaks them up into three parts so that the matched filter can run quicker
- processResults.py
  - *python processResults.py H1*
  - This file creates scatter plots using matched filter output files, plots the histogram of recovered SNR from running background matched filter, and the summary table of all results

Injection files

- The full files are found in injections/detector_cbc_injections.csv
- The files with _part0, _part1, _part2 added to the name are the first, second and third parts of the total data files

Matched Filter Output

- Found within matchedFilterResults/. The runMatchedFilter.py file is run three times for each detector, and thus, has three output files:
  - H1_cbc_injections_part0_matched.csv
  - H1_cbc_injections_part1_matched.csv
  - H1_cbc_injections_part2_matched.csv
- These files have been concatenated into a single file: H1.csv

Error Files

- Again, we have the three file per detector here
- Files logging the time of injection that failed to run for one reason or another
- Within the directory matchedFilterResults/ there are 5 directories that contain these injection times:
  - fileDNE/ - File did not exist when I ran it. See H1_DNE_error.csv
  - fileLen0/ - File had length of 0. See H1_len0_error.csv
  - injectionsNearEndFile/ - Injection was too near beginning or end of file to run. See H1_inj_nearEOF.csv
  - NaNinjection/ - NaNs in the data stopped injection from running. See H1_NaN_error.csv
  - shortFile/ - The file was under 12000 bytes. See H1_shortFile_error.csv (No errors of this type for H2)

Results

- Results/ directory contains plots and results/tables contains the final summary table

Appendix B: Result table (see attached file)

Column 1: GPS time
Column 2: Detector
Column 3: Mass 1 in solar mass units
Column 4: Mass 2 in solar mass units
Column 5: Distance in Mpc
Column 6: Status of injection, coded as:
1: Successful
2: Not in Science Mode
3: Injection process off
4: GRB ALERT;
5: GRB Alert and/or Operator override;
6: Injection Compromised
7: Not in Science Mode;
8: Injection process off
Column 7: Expected strength of template match
Column 8: Observed strength of template match
Column 9: Time difference (observed-expected) in seconds
Column 10: Comments