Undecided Title

Tony Lu

(Dated: July 8, 2016)

Abstract

waves coincidental?

The detection of gravitational was a piece of flash-news. In some sense, it was quite coincident that a detection was made almost immediately after the testing launch of Advanced Laser Interferometer Gravitational Wave Observatory. I took the waveform signal of the event GW150914 to make injections on data from LIGO's S6 run and recovered them. Overall, none of signal-to-noise ratio values I recovered was high enough to claim a detection. In the later portion of S6 where noise was less loud, the signal could be identified as a candidate event. However, I also noticed that if the distance from the system is closer, LIGO at S6 would have the potential to make the detection.

I. THE MAIN QUESTIONS

One of the deciding factors that made the detection of elusive gravitational waves possible is the sensitivity of the equipment. Over the LIGO's S5 [1], S6 [2] and O1 [3] (which the scientists used to refer to as S7) runs, the sensitivity was improved significantly each time, with effective attenuation of noise.

previously refered

At some degree, the detection seemed very much like a coincidence, for no detection in the past two decades was made until the one made within just one week after the testing launch of Advanced LIGO [4]. So here's a question: was LIGO at any early stage sensitive enough to make this detection? Was the detection of GW150914 a coincidence?

Good direct statement of your question.

II. BACKGROUND INFORMATION

More than a century ago Albert Einstein's General Theory of Relativity stated that we live in a four dimensional world—space-time (three dimensions for space and one for time). Masses cause distortions in spac-time, which accounts for the phenomenon of gravity. And the theory predicts that changing mass distribution will generally produce ripples in space-time, which is another prediction from him—gravitational waves. [5, 6] For decades, scientists have been struggling to detect gravitational waves from outer space. They upgraded their detectors over and over again, just in order to catch the whisper from distant celestial bodies. Fortunately, on September 14, 2015, the LIGO detectors successfully recorded the signals from two colliding black holes, which was the first direct detection of gravitational waves ever, a remarkable one. [3] It is labeled GW150914.

This section aims to provide background information about LIGO and gravitational waves. The section ...

A. Introduction to Gravitational Waves

Einstein found that his linearized weak-field equations have wave solutions. [3] These transverse waves are *ripples* in the *fabric* of space-time, caused by some strong and catastrophic happenings in the universe.

According to Einstein's calculations, gravitational waves are generated by objects accelerating with asymmetric motion. To be more precise, gravitational waves require changing

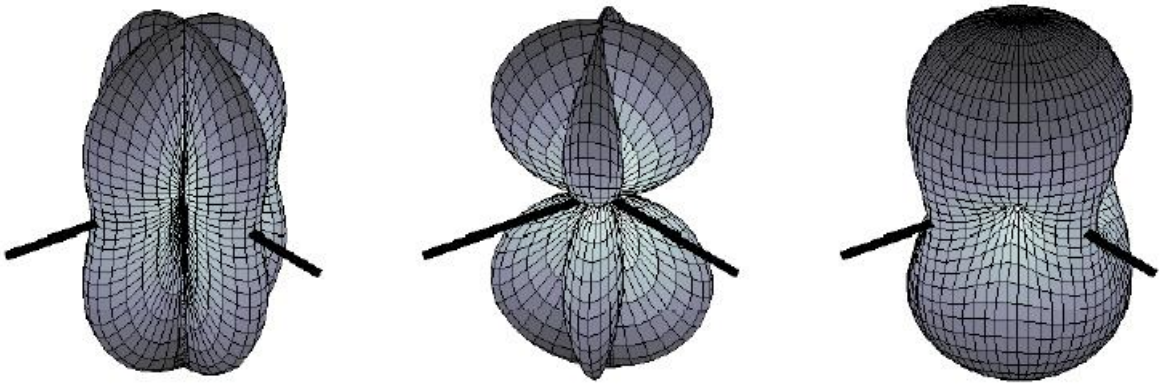

FIG. 1: The pattern on the left is for plus polarization, the middle pattern is for cross polarization, and the right-most one is for unpolarized waves. The black lines are the arms of LIGO detectors, which will be addressed in II B. Source: [8]

quadrupole mass distribution. Generally, the amount of gravitational waves given off is positively correlated to the system's mass and its speed of motion. Changes in space-time produced by a moving mass are not felt in the distance at once, but they propagate at the speed of light. [7]

Gravitational waves are transverse waves, the same as electromagnetic waves, but they are waves of changes in tensors (quadrupole distortions of space-time) which result in expansions and contractions in lengths in certain directions. Unlike the horizontal and vertical polarizations of electromagnetic waves, the those of gravitational waves are "plus" and "cross". However, gravitational waves do share lots of similarities with electromagnetic waves. They have frequencies and wavelengths, whose relationship is given by: $\lambda f = c$, where λ is the wavelength, f is the frequency, and c is the speed of light. They are able to carry energy, momentum, and angular momentum away from the source. [7] The strain amplitude can be derived from Einstein's quadrupole formula, and it's inversely proportional to the distance from the mass center. [6] It can also be measured by ratio of the change in length to the original length $h = \frac{\Delta L}{L}$. In text you can use $h = \Delta/L$ instead of a fraction.

Gravitational waves have various sources, but there are mainly four catagories: stochastic backgroud, bursts from gravitational collapse, pulsars, and binary systems. Binary systems refer to those that consist of two bodies rotating aroung each other. Whether or not the two stars collide, gravitaional waves will be generated. The detection made on September 14,

2015 is the collision of two black holes, which is named compact binary coalescence—"chirp". [3] More details are given by [7] in section 3.

B. Introduction to LIGO

Distortions in space-time is almost impossible to measure directly, for if you use a meter stick, the meter stick itself will expand and contract with changing space-time. Fortunately, lights remain the same speed despite any deformation of space-time, which sets the basis for the LIGO detectors. [9]

The detectors of LIGO—Michelson Interferometer—was invented by Albert Abraham Michelson, first used in the famous Michelson-Morley experiment [10]. Its function is for optical interferometry. With a beam splitter, a light source is split into the two arms, and each of the new beam is reflected back and combined again. The result of such combination leads to optical interferenceeither constructive or destructivedepending on the changes in length of the arms. Therefore, the photo detector could sense the variation in the arms length. [9]

The interferometer in LIGO consists of two vacuum arms of equal length—4 kilometers long, which form an L shape. There is a laser light source and a photodetector at about the corner of the L, and a beam splitter at the corner. At each end and in the middle of the arms freely hanged four mirrors which reflects the laser beam, and the two beams are combined into one at the beam splitter. Those mirrors in the middle are used to increase the distance light beams travel, and as a result, the actual effective length is increased from 4km to 1120 km, greatly enhancing LIGO's sensitivity. If the arm length changed, the two light beams which were originally in phase would be out of phase, creating destructive interference which allows the photodetector to sense the infinitesimal change in length of the arms. [9, 11]

Moreover, the two LIGOs in the US are located in Livingston Parish , Louisiana and Hanford , Washington. Both are L-shaped and of the same size–their arms are all 4km long. The two Ls are in different directions, and they are separated by 3030.3 kilometers on land, which means the planes these observatories are on have a dihedral angle of 27.3 degrees. [28] Therefore, it's guaranteed that gravitational waves from any directions can be detected, given enough sensitivity.

A brief history of LIGO is given by [12]. A more technically and quantatatively detailed

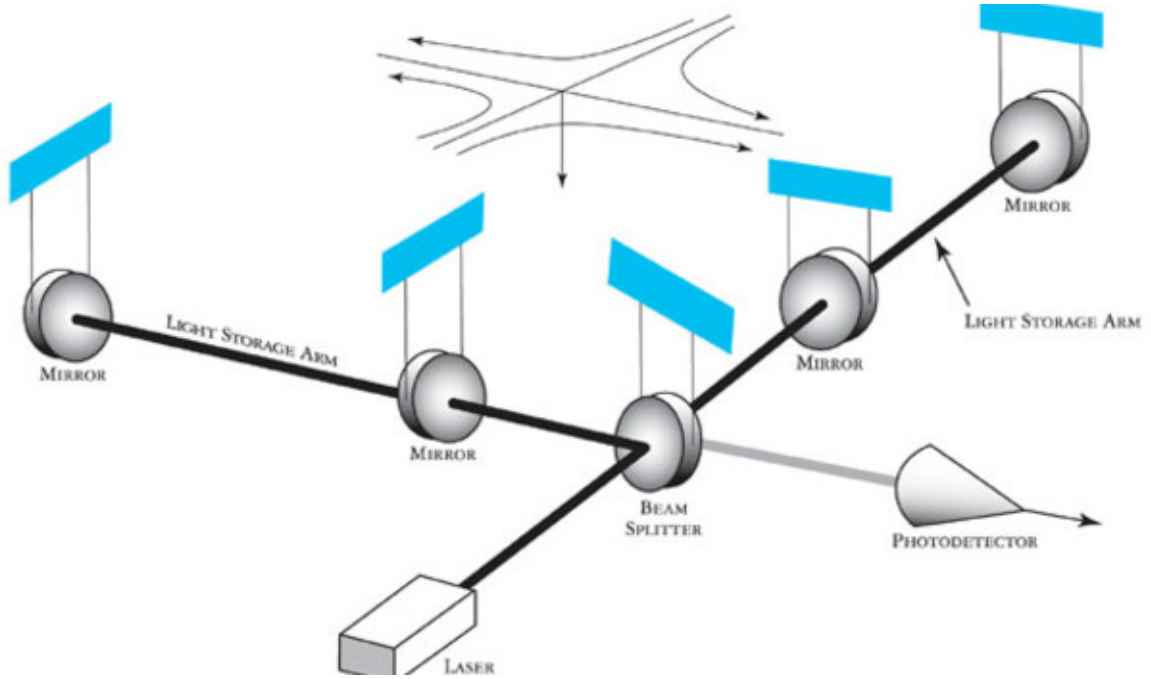

FIG. 2: Basic schematic of LIGO's interferometers with an incoming gravitational wave depicted as arriving from directly above the detector. Source: [9]

description of LIGO is given by [8].

C. Sensitivity of the Detectors

Similar to any real physics experiment, the LIGO detectors face various disturbance from the environment. Considering the small amplitudes of gravitational waves, we see that sensitivity become decisive. At the very beginning of the construction, the interference of the noises are so high that it was even impossible for the detectors to collect data clean enough to analyze. But the quality of data collected has been significantly improved over the several stages of LIGO. The datasets of its S5 (2005-2007), S6 (2009-2010), and O1 (2015-present) are available online. [13] Among all these sensitivity-limiting factors are primarily: seismic noise (at relatively low frequencies), thermal noise (at mid frequencies), shot noise (at high frequencies), and the problems with the lasers and electronics. Some more specific details, including solutions, are given by [14–16]

The problems related to sensitivity are curious to investigate. My research is closely related to sensitivity.

September 2016

O1 has ended.
It think it ended in Jan 2016.
Check.
And O1 data are not all available yet, just the selected events.

D. Injections

In short, injections are used to test if the detectors are working well. There are two types of injections: hardware injections and software injections. The first type is made by manually changing the position of the mirrors in the interferometer. The second type is achieved by adding signal directly into the data flow. Moreover, there were those called "blind injections", when few scientists know it was an injection while the majority were convinced that it really was a detection. One of the famous events was the one back in 2010, which was reported in [17].

E. The GPS Time

GPS time is a system to keep track of time. It is the total number of seconds from 00:00:00 January 6, 1980 UTC.

III. RESEARCH METHODOLOGY DESCRIPTION

This section aims to provide a description of the overall methodology to obtain the raw data for further analysis of my investigation.

A. Data Files

1. Event Waveform

The strain data of the GW150914 event I used is from the LOSC site. Specifically, it's form the waveform.txt file from the "Download the Data" section in [18]. This strain data was later used as the signal for my self-made injection.

I try to use a courier for file names to show it's a "computer thing"

2. Template for Recovery

As for the template for injection recovery (see: III D), it's different from the waveform. It is included in the zip file of another tutorial. [19] The complex template contains two waveforms, one of which is used as the real part and the other as the imaginary part.

So you used waveform.txt for the injection, but the HDF5 template for the recovery?

Moreover, as pointed out in the "Waveform Template" of this tutorial, the templates are not 100% the same as what the scientists actually used. Many subtleties are skipped, for example. But the quality of the templates online is good enough for my investigation.

3. Background—Noise

The background strain data of my injections were selected from a number of data files from LIGO's S6 run, from GPS time 931035615 to 971622015. The S6 data archive is on this page [20]. Because of the probable fluctuation in the power of the background noise over the entire S6 run, I chose 10 different and relatively evenly dispersed data files, each starting at GPS time 931127296 (0.226%), 934846464 (9.39%), 941707264 (26.3%), 941785088 (26.5%), 947154944 (39.7%), 952623104 (53.2%), 959344640 (69.8%), 963629056 (80.3%), 967442432 (89.7%), and 971407360 (99.5%), all lasting for 4096 seconds.

Try more descriptive name/indicator.

B. Data Quality Vetoes

The background quality is critical. A piece of low-quality data could result in a misleading outcome. Furthermore, that kind of datafiles would be vetoed by the scientists. Even if there really were an event, it would not be searched and detected. And it becomes useless for me do investigate on such a data file.

In general, I needed to assure that the background I was injecting on has its proper data quality flags on. Because the event of GW150914 is a compact binary coalescence system [3] and the mass is big enough to be catagorized as "high mass" [21], the flag of "CBCHIGH_CAT4" should be on. Besides, there should not be any other injections which would affect my recovery, so the flag of "HW" should be off. [22]

C. Making the Injections

The injections were made by simply superposing the waveform on to the background signal. Ten injections were made to each of the ten data files at randomly selected points. The data quality flags for compact binary coalescence hardware injections were modified as well. In addition to the original strain, I amplified the waveform signal by a factor of 2, 3,

5, 8, 12, 20, 40, and zero, and thus did 8 more groups. In total, that was 800 injections. The script for making these injections is attached in VII A.

D. Injection Recovery

To recover the injection, I computed the SNR of each injection using matched filter. The processing was essentially the same as that in [18]. The scripts for recovery are attached in VII B.

E. Software Support

The main software for analysis is Python 2.7.11, Anaconda 2 (iPython), and Excel 2013 with excel2latex macro.

F. Some Relevant Statistical Concepts and Algorithms

1. Signal-to-noise Ratio—SNR

Signal-to-noise ratio (SNR) [23] indicates the relative power of the signal to the background noise. The value is a positive figure greater or equal to one.

2. Matched Filter

Matched filter [24] is an algorithm that computes the SNR.

3. Root Mean Square—RMS

Root mean square is calculated by taking the square root of mean of the squared values of all the samples: $x_{RMS} = \sqrt{\frac{x_1^2+x_2^2+...+x_n^2}{n}}$ [25]

Flesh this all in as text rather than outline form (we discussed this in our meeting).

4. Linear Regression

Linear Regression is an approach to model the relationship between a dependent variable and an independent variable. The result is that the depend variable is a linear function of

the independent one: $y = ax + b$ [26]

Additionally, there is a concept called correlation coefficient. The result, r^2, is a measure of how well the values fit the model, therefore indicating the accuracy of the model. The closer r^2 is to 1, the better the values fit the model. If r^2 equals one, all the values lie on the model. [27]

IV. DATA ANALYSIS

This section presents the data analyzing process. The data files obtained in III A 3 are referred to as data file number one to ten, respectively.

A. Computing the RMS of Strain Data

In this case, RMS functioned as a representation of the significance of the noise in each data file. It can be inferred from the RMSs that noise in the detectors is generally louder earlier in S6 than later in S6, which is consistent with other results coming up.

##Insert Scatterplot Graph##

B. The Mean SNR of Each Data File at Different Amplifications

Before calculating the mean SNR of each piece of data, I manually discarded some extreme values which appeared to be anomalies. These extremities may be caused by a sudden fluctuation in the background noise, a mistake in the computation, or an injection failure. The mean SNR was simply taken by computing the average recovered SNR value of the 10 injections (except the anomolies) of each data file at each amplification.

The figures in Table I indicate that the matched filter output for background noise is generally at a level of 4 or 5. Sometimes, there may even be a spike in the SNR that reaches 6 or more. So the SNR values taken where the injection signals were not amplified (aplification equals one) are not credible, for they are mostly 6 or less. Instead of the injections, those SNRs could have been caused the spikes in the background noise. Therefore, I needed to use regression analysis to model the SNR against amplification, thus predicting the SNR where the signal was not amplified.

TABLE I: The average SNRs recovered from the ten data files at 8 different amplifications of the injection signal. The amplification of zero means the SNR is recovered from pure background noise.

which represents a false detection (there is no signal)

	Amplification								
File Number	0	1	2	3	5	8	12	20	40
1	4.967	4.963	8.053	12.886	21.483	32.060	49.788	79.564	148.866
2	4.864	4.851	8.947	13.161	20.177	35.378	51.638	74.457	154.196
3	6.235	6.184	7.973	12.590	20.627	33.050	49.999	80.095	150.797
4	5.587	5.766	8.664	14.936	26.684	40.671	60.653	94.043	175.344
5	4.626	5.096	10.326	15.288	25.616	40.149	61.203	98.755	175.360
6	4.613	6.160	11.861	18.152	29.805	46.423	70.410	116.439	206.434
7	5.100	8.176	12.717	18.842	31.322	49.614	70.877	118.420	202.897
8	4.675	6.032	12.165	17.230	34.306	46.488	70.337	111.888	191.332
9	4.570	6.479	10.079	17.501	28.178	47.082	64.857	113.912	203.616
10	4.980	6.489	12.096	19.185	30.698	50.930	67.298	120.715	206.301

C. Computing the Linear Regression

Can you plot these on a single graph, with different markers for each file, so we can see visually that they are linear? If not all, then one as a representative.

First of all, I chose linear regression to be the type of my model. Due to the nature of the concept of SNR, the amplification (signal power) should be positive correlated to the output. Meanwhile, the output goes up neither exponentially nor logarithmically, but it increases at a steady rate, so linear model should be the best fit. I further specified the intercept as zero, for theoretically the output should be zero when no signal exist in the noisy background.

If we were to test the models, we would find that the regressions would somewhat underestimate the SNRs at low amplifications. So in reality, the SNRs should be at somewhere between the second colume of data in Table I and the values in Table II. Nevertheless, the highest possible SNR was only around 6.5, which means the signal would be overwhelmed by the noise and unable to stand out.

I'm interested to see how the graph intercepts 1 not 0, based on a straight line from higher values. The 0 value is the background, and the projected value at 1 is what you extrapolate as the detectable (or not) signal.

TABLE II: Linear regression computed from the data in Table I, with intercept set to zero. The leftmost colume presents the predicted SNR when no amplification was made.

File Number	Slope	R-Squared	Predicted SNR
1	3.812		3.812
2	3.882		3.882
3	3.854		3.854
4	4.516		4.516
5	4.559		4.559
6	5.356		5.356
7	5.330		5.330
8	5.055		5.055
9	5.250		5.250
10	5.395		5.395

I'm not sure these values make sense. It's the amplification of 1 not zero that represents the actual signal SNR. Again, one might just "see" that from the graph above and you would not need the table. General rule: use a graph instead of a table, unless you can't.

V. CONCLUSION

I found that the GW150914 event would not be loud enough for LIGO at S6 run. The signal could, at best, be identified as a candidate event, but it's significance would be far from enough to claim a real detection. In short, the S6 run is not sensitive enough to make this detection. Therefore, GW150914 event was not a mere coincidence. In contrast, the successful detection was and (the future detections are) made possible by persisting in upgrading the detectors, which are really worth doing.

was

VI. EXTENSIONS

VII. APPENDIX A: PYTHON SCRIPTS

Put your list of references before the computer code appendix.

These are the Python scripts I used for my research.

A. For Making Injections

You can start this on a new page.
You could also put two pages on one for code,
(You might have to merge 2 PDF's for that.)

```
# Programmer: Zhehao Lu Tony

September 2016
```

Suggest you use a fixed-width font such as Courier for computer code.
A smaller font is okay for code.

Grade: 8/10

```
# 2AP2, International Curriculum Center,
# Shenzhen Foreign Languages School, China
# Date: June 28, 2016
# For making software injections on raw LIGO signal data.
# Recommend make a copy of the original file and then do the injection,
# because the injections cannot be deleted easily.
# One can make injections on as many files one time as he/she wants
# by altering the 'fileName[]' array.

import h5py
import numpy as np
import matplotlib.pyplot as plt
import readligo as rl
from random import randint

#--- Sampling rate equals to 4096 Hz
fs = 4096

#--- Read in the waveform
temp_time, temp = np.genfromtxt('GW150914_4_NR_waveform.txt')\
                  .transpose()

#--- Set the amplification
print 'Please input the amplification of the signal.'
amplification = input()
temp *= amplification

#--- Define the function to segment the segments into suitable segments
def slice_filter(dq, hw):
    #--- Cut the whole segment in to small ones of 60s, with spacing of 1s
    i = 0
```

```
    while i < len(dq):
        if dq[i].stop - dq[i].start < 60*fs:
            del dq[i]
            continue
        dq.insert(i, slice(dq[i].start, dq[i].start + 60*fs))
        dq[i+1] = slice(dq[i].stop + 1*fs, dq[i+1].stop)
        i += 1
    #-- Delete the segments with injection activated
    i = 0
    j = 0
    if len(hw) == 0:
        return dq
    while i < len(dq) and j < len(hw):
        #-- dq is 100% before hw
        if dq[i].stop < hw[j].start:
            i += 1
        #-- dq is 100% after hw
        elif dq[i].start > hw[j].stop:
            j += 1
        #-- dq and hw overlap
        else:
            del dq[i]

    return dq

#-- Prepare the files to load
fileName = []
fileName.append('H-H1_LOSC_4_V1-931127296-4096')
fileName.append('H-H1_LOSC_4_V1-934846464-4096')
fileName.append('H-H1_LOSC_4_V1-941707264-4096')
fileName.append('H-H1_LOSC_4_V1-941785088-4096')
```

```
fileName.append('H-H1_LOSC_4_V1-947154944-4096')
fileName.append('H-H1_LOSC_4_V1-952623104-4096')
fileName.append('H-H1_LOSC_4_V1-959344640-4096')
fileName.append('H-H1_LOSC_4_V1-963629056-4096')
fileName.append('H-H1_LOSC_4_V1-967442432-4096')
fileName.append('H-H1_LOSC_4_V1-971407360-4096')

#-- Prepare the document to record the injections
f = open('Injection_M_{0}.txt'.format(amplification), 'w')

#============================#
#=======Main Part============#
#============================#
for name in fileName:

    inj_samp = []

    #-- Load the original file
    rawFile = h5py.File(name + '.hdf5', 'r')
    strain_raw = rawFile['strain/Strain'].value
    dq_raw = rawFile['quality/injections/Injmask'].value
    CBCHIGH_CAT4 = (rawFile['quality/simple/DQmask'].value >> 4) & 1
    HW_CBC = (rawFile['quality/injections/Injmask'].value >> 0) & 1
    GPSstart = rawFile['meta/GPSstart'].value
    rawFile.close()

    #-- Load the file again which will be injected
    dataFile = h5py.File(name + '.hdf5', 'r+')
    strain = dataFile['strain/Strain']
    dqInj = dataFile['quality/injections/Injmask']
```

```
#-- Getting the segement lists
segList = rl.dq_channel_to_seglist(CBCHIGH_CAT4)
segList_HW = rl.dq_channel_to_seglist(HW_CBC)
segList = slice_filter(segList, segList_HW)

#-- Pick 10 (or if not more than 10, all) segments randomly
inj_num = []
if len(segList) <= 10:
    inj_num = range(0, len(segList))
    #-- If less than 10, throw a caution
    print 'Caution: there are less than ten suitable spots in %s!'\
                % name
    print '{0} only!'.format(len(segList))
else:
    while len(inj_num) < 10:
        i = randint(0, len(segList)-1)
        if i in inj_num:
            continue
        inj_num.append(i)
    inj_num.sort()

#-- Make the injection to every piece
for i in inj_num:
    seg = segList[i]
    #-- Get a random starting point
    inj_sample = randint(seg.start, seg.stop - temp.size)
    inj_samp.append(inj_sample)
    #-- Superpose the waveform to the signal
    for j in range(0, temp.size):
        strain[inj_sample + j] += temp[j]
    #-- Write onto dq flag: HW_CBC
```

```
            Start = seg.start / fs
            End = seg.stop / fs
            for j in range(Start,End):
                dqInj[j] = (1 << 0) | dqInj[j]
                dqInj[j] = (1 << 1) | dqInj[j]

        #-- Record the injections
        for sample in inj_samp:
            time = sample * 1. / fs
            f.write('{0} {1} {2} {3}\n'.format(GPSstart, GPSstart+time,\
                                               time, sample))
        print 'File %s injection at M={0} succeed.'.format(amplification)\
                                              % name

        #-- Close the file
        dataFile.flush()
        dataFile.close()

f.close()
```

B. For Recovery—Bulk Operation

```
# Programmer: Zhehao Lu Tony
# 2AP2, International Curriculum Center,
# Shenzhen Foreign Languages School(Shenzhen, China)
# Date: June 30, 2016
# For recovering the self-made injections in raw LIGO data



import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
```

```
import scipy.signal as sig
import readligo as rl
import h5py

#--- Read in the template
f_template = h5py.File('GW150914_4_template.hdf5', "r")
template_p, template_c = f_template['template'].value
temp_strain = template_p + template_c * 1.j

#--- Read in the names
fileName = []
fileName.append('H-H1_LOSC_4_V1-931127296-4096')
fileName.append('H-H1_LOSC_4_V1-934846464-4096')
fileName.append('H-H1_LOSC_4_V1-941707264-4096')
fileName.append('H-H1_LOSC_4_V1-941785088-4096')
fileName.append('H-H1_LOSC_4_V1-947154944-4096')
fileName.append('H-H1_LOSC_4_V1-952623104-4096')
fileName.append('H-H1_LOSC_4_V1-959344640-4096')
fileName.append('H-H1_LOSC_4_V1-963629056-4096')
fileName.append('H-H1_LOSC_4_V1-967442432-4096')
fileName.append('H-H1_LOSC_4_V1-971407360-4096')

#--- Prepare the document to record the SNRs
print 'For what level of amplification are you recovering?'
amplification = input()
f = open('SNR_M={0}.txt'.format(amplification), 'w')

for Name in fileName:
    print Name + ':'
    #--- Load the data
    fileName = Name + '.hdf5'
```

```
raw_strain, raw_time, dq = rl.loaddata(fileName)
fs = int(1. / (raw_time[1] - raw_time[0]))

#-- Prepare the segment list
segList = rl.dq_channel_to_seglist(dq['HW_CBC'])

#-- Prepare the txt file to record the result
filename = Name + '.txt'

for seg in segList:

    #-- If not self-injection, skip it
    if seg.stop - seg.start == 30:
        continue
    strain = raw_strain[seg]
    time = raw_time[seg]

    #-- Compute the PSD
    NFFT = fs * 4
    NOVL = NFFT / 2
    psd_window = np.blackman(NFFT)
    Pxx, fpsd = mlab.psd(raw_strain, Fs=fs, NFFT=NFFT, \
                         window=psd_window, noverlap = NOVL)
    Pxx_temp, fpsd_temp = mlab.psd(temp_strain, Fs=fs, NFFT=NFFT)

    #-- Optimal matched filter
    dwindow = sig.tukey(strain.size, alpha = 1./8)
    strain_fft = np.fft.fft(strain * dwindow) / fs
    temp_padded = np.append(temp_strain,\
        np.zeros(strain.size - temp_strain.size))
    dwindow2 = sig.tukey(temp_padded.size, alpha = 1./8)
```

```
    temp_fft = np.fft.fft(temp_padded * dwindow2)
    seg_freq = np.fft.fftfreq(strain.size) * fs
    #Interpolate to get the PSD values at the needed frequencies
    power_vec = np.interp(np.abs(seg_freq), fpsd, Pxx)
    #Op.Ma.Fi.:
    optimal = strain_fft * temp_fft.conjugate() / power_vec
    optimal_time = 2 * np.fft.ifft(optimal) * fs
    #Normalize:
    df = np.abs(seg_freq[1] - seg_freq[0])
    sigmasq = 1 * (temp_fft * temp_fft.conjugate() / \
                   power_vec).sum() * df
    sigma = np.sqrt(np.abs(sigmasq))
    SNR = abs(optimal_time / sigma)

    #-- Print the result
    Max = SNR.max()
    found_time = time[np.where(Max == SNR)] + 15
    print 'Recovered SNR: {0} at {1}'.format(Max,found_time[0])

    #-- Record the result
    f.write('{0} {1} {2}\n'.format(int(raw_time[0]),\
                                   found_time[0], Max))

f.close()
```

VIII. APPENDIX B: MORE DATA TABLES

I hope not.

---

[1] LIGO Scientific Collaboration, *The s5 data release*, URL `https://losc.ligo.org/S5/`.

[2] LIGO Scientific Collaboration, *The s6 data release*, URL `https://losc.ligo.org/S6/`.

[3] B. P. Abbott et al., Phys. Rev. Lett. (2016).

[4] LIGO CalTech, *Ligo timeline*, URL https://www.ligo.caltech.edu/page/timeline.

[5] A. Einstein, Sitzungsber. K. Preuss. Akad. Wiss **1**, 688 (1916).

[6] A. Einstein, Sitzungsber. K. Preuss. Akad. Wiss **1**, 154 (1918).

[7] B. Sathyaprakash and B. F. Schutz, Living Rev. Relativity **12** (2009), URL http://www.livingreviews.org/lrr-2009-2.

[8] B. P. Abbott et al., Rept. Prog. Phys. **72** (2009).

[9] LIGO Caltech, *What is an interferometer?*, URL https://www.ligo.caltech.edu/page/what-is-interferometer.

[10] A. Michelson and E. Morley, American Journal of Science pp. 333–345 (1887).

[11] MIT LIGO, *A comprehensive overview of advanced ligo* (2015), URL https://www.advancedligo.mit.edu/summary.html.

[12] K. Thorne and R. Weiss, *A brief history of ligo* (2016), URL https://www.caltech.edu/content/brief-history-ligo.

[13] LIGO Scientific Collaboration, *Timeline*, URL https://losc.ligo.org/timeline/.

[14] M. Pitkin, S. Reid, S. Rowan, and J. Hough, Living Rev. Relativity **14** (2011).

[15] Y. Levin, Phys. Rev. D **57**, 659 (1998).

[16] D. M. Macleod, S. Fairhurst, B. Hughey, A. P. Lundgren, L. Pekowsky, J. Rollins, and J. R. Smith, Class. Quant. Grav. **29** (2012).

[17] LIGO Scientific Collaboration, *Blind injection: Stress-tests ligo and virgo's search for gravitational waves*, URL http://www.ligo.org/news/blind-injection.php.

[18] LIGO Scientific Collaboration, *Signal processing with gw150914 open data* (2016), URL https://losc.ligo.org/s/events/GW150914/GW150914_tutorial.html.

[19] LIGO Scientific Collaboration, *Binary black hole signals in ligo open data* (2016), URL https://losc.ligo.org/s/events/GW150914/LOSC_Event_tutorial_GW150914.html.

[20] LIGO Scientific Collaboration, *S6 data archive*, URL https://losc.ligo.org/archive/links/S6/H1/931035615/971622015/simple/.

[21] J. Aasi, Phys. Rev. D **87**, 022002 (2013).

[22] LIGO Scientific Collaboration, URL https://losc.ligo.org/archive/dataset/S6/.

[23] D. H. Johnson.

[24] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, Phys. Rev. D **85**, 122006 (2012).

No need to say "URL"

[25] E. W. Weisstein, *Root-mean-square*, URL `http://mathworld.wolfram.com/Root-Mean-Square.html`.

[26] E. W. Weisstein, *Linear regression*, URL `http://mathworld.wolfram.com/LinearRegression.html`.

[27] E. W. Weisstein, *Correlation coefficient*, URL `http://mathworld.wolfram.com/CorrelationCoefficient.html`.

[28] By Google Earth ??? more detail?