LIGO Open Science Center Workshop

Jonah Kanner, Branson Stephens, Michele Vallisneri, Alan Weinstein, Roy Williams

LIGO-G1400509v4

LIGO

Why LOSC?

- Access data with ease
  - No need to login to LDAS cluster
  - Point and click download of LIGO data
- Complete documentation and tutorials
  - Easy start-up for new students
  - Everything we do today (and more!) in online tutorials
- Easy access to segment and DQ information
- Load data in GW Frame or HDF5 format
  - LSC software stack is optional
  - Compatible with Python, Matlab, C, and many others

LIGO

Step 0: Software Setup

- Workshop assumes you have installed python, along with h5py, numpy, matplotlib
- Software set-up instruction is at:
  - https://losc.ligo.org/tutorial00/
- Create a directory for this workshop
  - Let's call it "workshop"
  - Download the LOSC API, readligo.py, to the workshop directory:
    https://losc.ligo.org/s/sample_code/readligo.py

LIGO

Step 1: Find data with "Timeline"

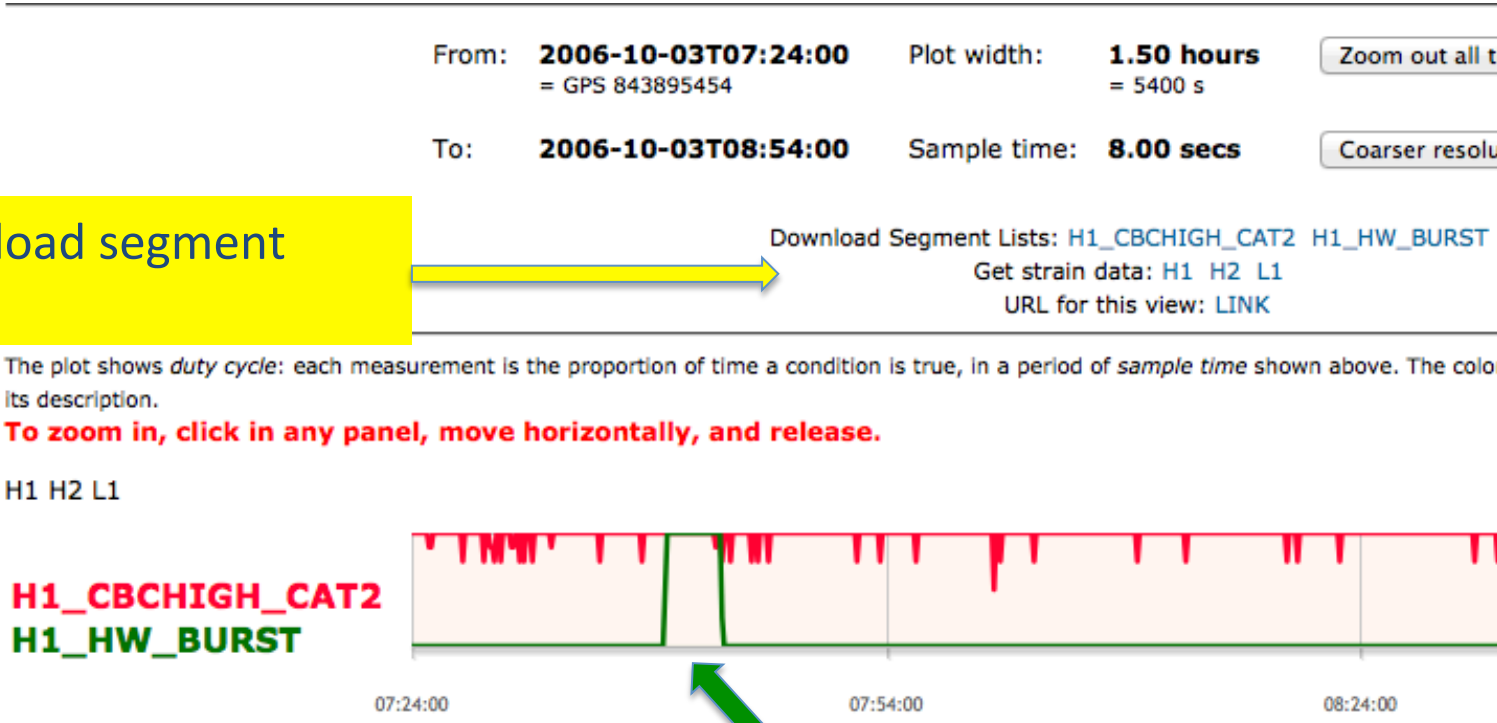
- Timeline contains segment information
  - DATA available, Data Quality, and injections
- Timeline can be used to:
  - Find times of good (or bad) data quality
  - Learn when GW detectors were running
  - Download segment lists
  - Download LIGO data files

LIGO

Step 1: Find data with "Timeline"

- Go to the S5 Data Page:
  - https://losc.ligo.org/S5/
- Click "S5 Timeline Query Form" near the bottom
  - Enter a start time of 843895500
  - Enter a duration of 3600
  - Choose the **Plot** option
  - Choose the H1_CBCHIGH_CAT2 and H1_HW_BURST flags
  - Click the link at the bottom to see the timeline

Step 1: Find data with "Timeline"



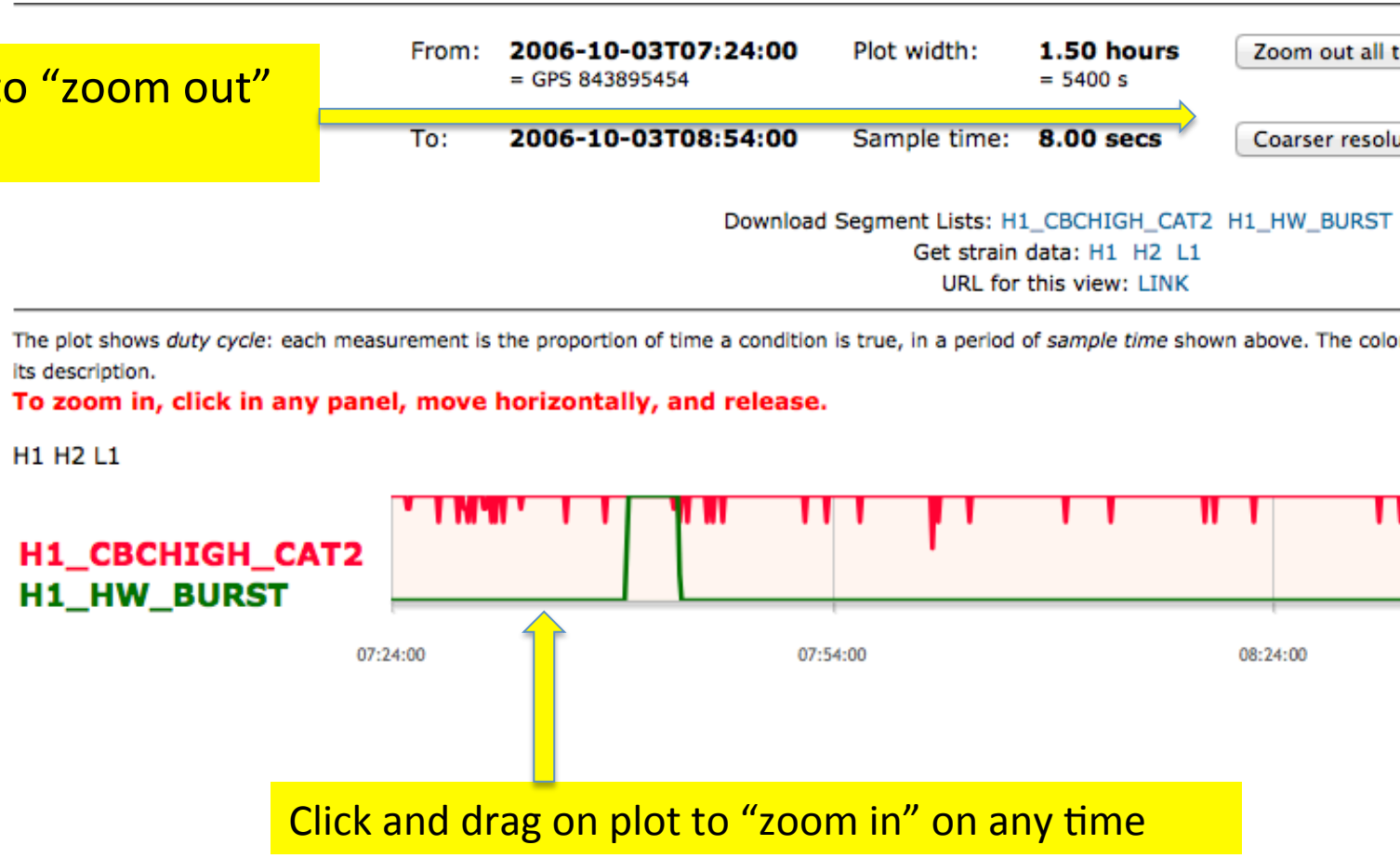
Links to download segment lists and data

The red curve shows that this data "passes" CAT 2 flags for the CBC High mass search most, but not all, of the time.

The green curve shows the time of a burst hardware injection

Step 1: Find data with "Timeline"

Can use buttons to "zoom out" to see all of S5



Click and drag on plot to "zoom in" on any time

LIGO

Step 2: Download data

- From the timeline page, click the link for "Get the strain data" for "H1"

Timeline	UTC	Mbytes	HDF5	Frame	Percent
843894784	2006-10-03T07:12:50	129 MB	HDF5	Frame	100
843898880	2006-10-03T08:21:06	129 MB	HDF5	Frame	100

- Each line represents 4096 s of LIGO data
- Click the links that say "HDF5" to download the files
  - Notice GW frame, or .GWF, format is also available
- Save the files to your "workshop" directory or a sub directory
  - Directions for automated downloads are in the tutorials

LIGO

Segment Lists

- Most data analysis projects start with a segment list
  - May be times of SCIENCE mode data
  - May also be times of passing some data quality level (For example, BURST_CAT2)
- LOSC provides segment information in 2 ways:
  - Segment lists may be downloaded from web through "Timeline" (See Slide 6)

OR

  - Segment lists may be constructed from bit masks in LOSC data files

LIGO

Step 3: Create a segment list

- Start python (or iPython, or Canopy, or Anaconda, or an iPython notebook)
- Make sure your working directory is the 'workshop' directory you created
- Enter commands as a script in a text file
  - Not directly in the interpreter
- Import the modules we'll use:

```
import readligo as rl
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
plt.ion()
```

LIGO

Step 3: Create a segment list

- Define the times that you have downloaded data.

```
dstart = 843894784
dstop = dstart + 4096
```

- Use the LOSC API to create a segment list

```
# -- Create a CBC High mass post CAT 2 segment list
seglist = rl.getsegs(dstart, dstop, 'H1',flag='CBCHIGH_CAT2')
print seglist
```

Note: The data for the segment lists are contained in the files you already downloaded. There are no calls to a segment data base.

LIGO

Step 4: Load LIGO data

- We'll write a loop over each segment in the list we just made

```
# -- Load the data for each of the first 3 segments
for (start, stop) in seglist[0:3]:
    strain, meta, dq = rl.getstrain(start, stop, 'H1')
```

- The variable "strain" now contains this segment of data as a numpy array
- You got the data. That's it!
  - But, we'll work a few more examples just for fun ☺

LIGO

Step 4: Load LIGO data

- Now that we've got a loop to load data, let's add a few lines to plot an ASD for each segment

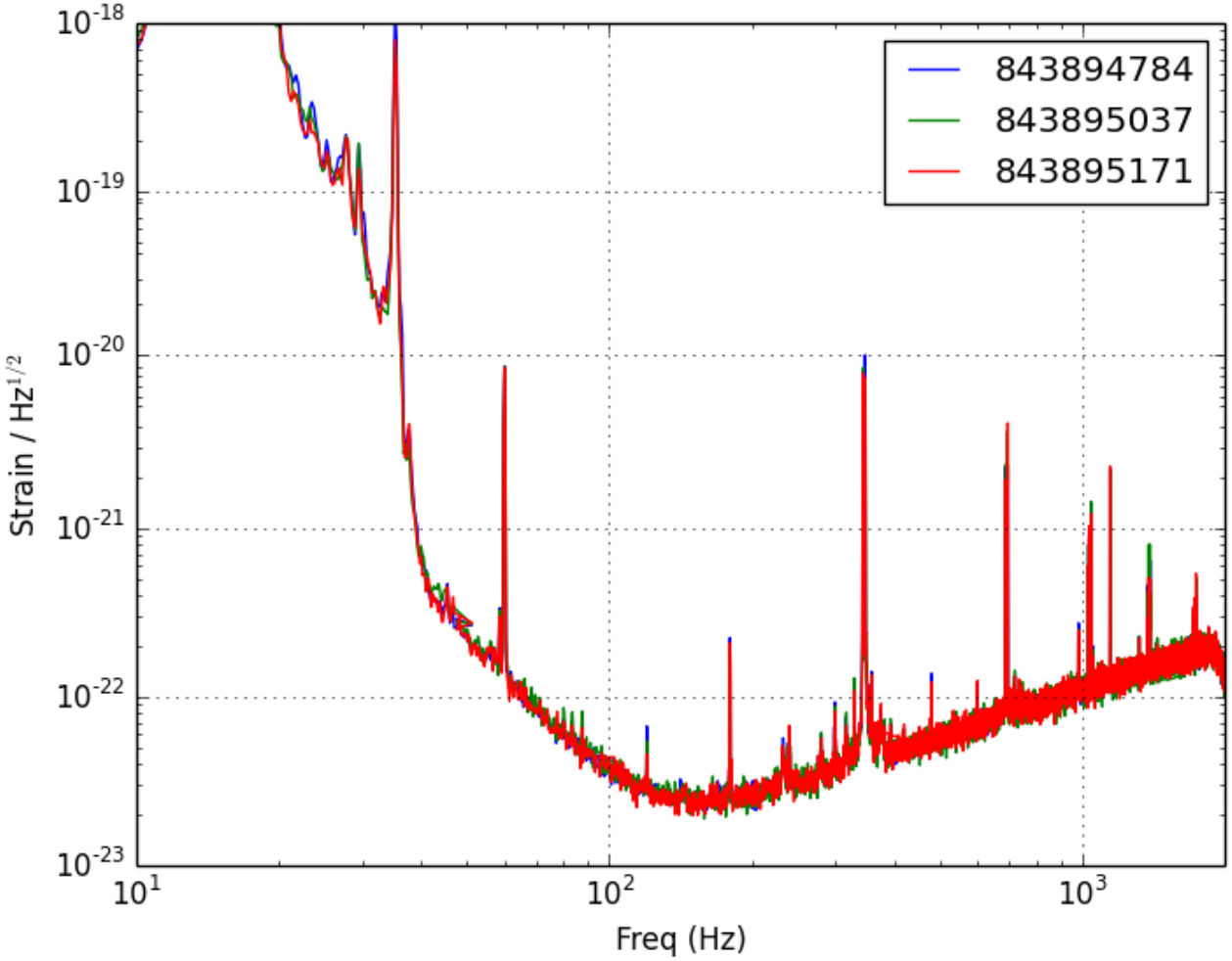
```
# -- Load the data for each of the first 3 segments
for (start, stop) in seglist[0:3]:
    strain, meta, dq = rl.getstrain(start, stop, 'H1')

    #-- For each segment, plot the ASD
    fs = int(1.0/meta['dt'])
    Pxx, freqs = mlab.psd(strain, Fs = fs, NFFT=4*fs)
    plt.loglog(freqs, np.sqrt(Pxx), label=str(start))

# -- Make the plot look nice
plt.axis([10, 2000, 1e-23, 1e-18])
plt.grid('on')
plt.xlabel('Freq (Hz)')
plt.ylabel('Strain / Hz$^{1/2}$')
plt.legend()
```

LIGO

Step 4: Load LIGO data



LIGO

Hardware Injection Documentation

- The LOSC web site includes hardware injection documentation
  - Burst, CW, Stochastic, and CBC
  - Includes detailed lists of injection times & parameters
  - Hardware Injection times are marked in data quality segments
- Tutorial with complete instructions for finding a CBC hardware injection
  - https://losc.ligo.org/tutorials/

LIGO

Find a hardware injection

- We can use the LOSC segment information to find times of hardware injections
  - The full list of S5 DQ flags is at:

    https://losc.ligo.org/archive/dataset/S5/

```
# -- Define a segment list of hardware injection times
seglist = rl.getsegs(dstart, dstop, 'H1', flag='HW')
print seglist
```

LIGO

Find a hardware injection

- This is a burst hardware injection, so maybe it will show up in a spectrogram. We can load the data the same way we did earlier.

```
# -- As before, load the data for each segment
for (start, stop) in seglist:
    strain, meta, dq = rl.getstrain(start, stop, 'H1')

    # -- Make a spectragram of the data
    fs = int(1.0/meta['dt'])
    NFFT = fs/2
    window = np.blackman(NFFT)
    spec_power, freqs, bins = mlab.specgram(strain, NFFT=NFFT, Fs=fs,
                                            window=window)
```

LIGO

Find a hardware injection

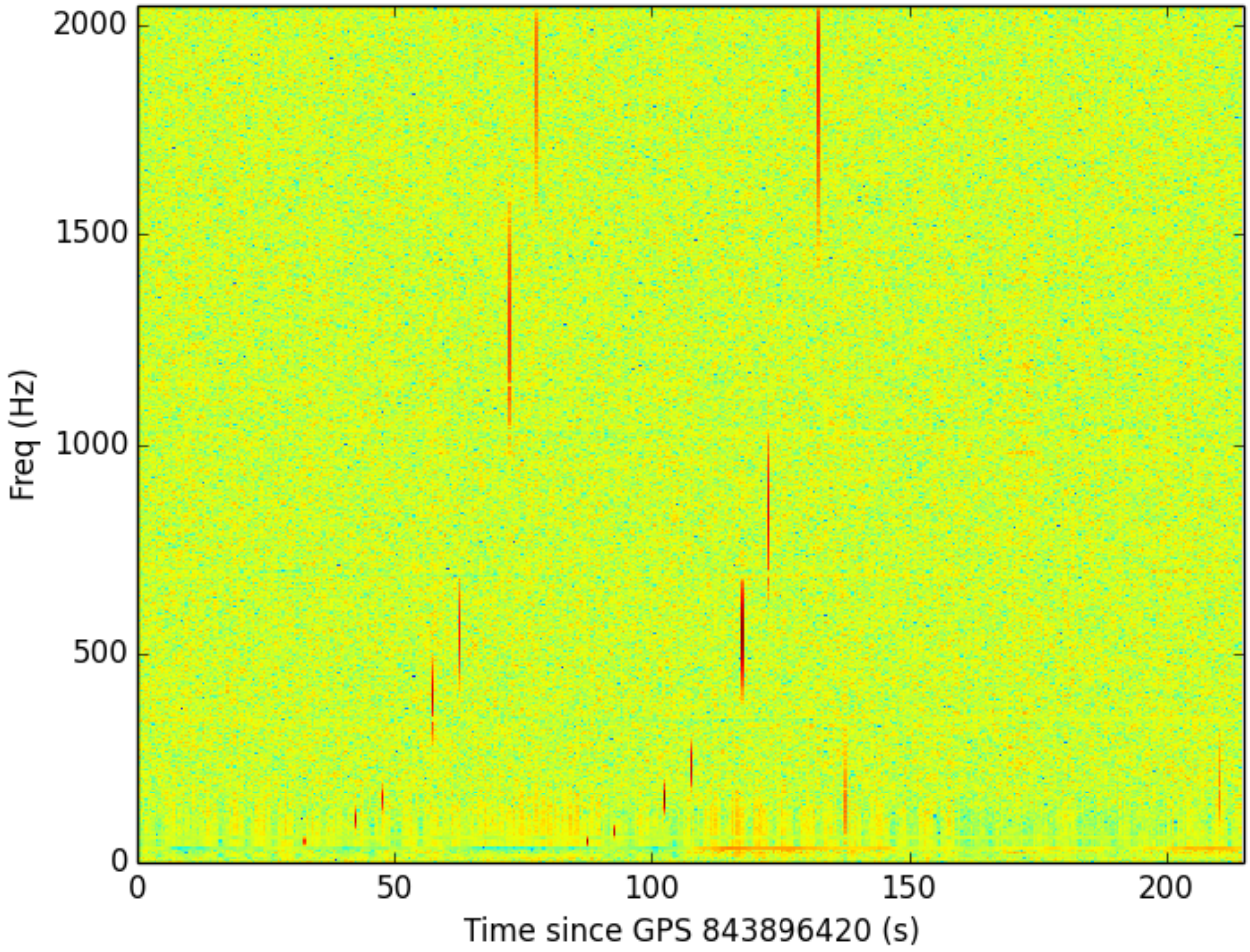
- Before plotting the spectrogram, we'll normalize each frequency bin by its median power

```
# -- Normalize each frequency bin to the median power in that bin
for n in range(spec_power.shape[0]):
    spec_power[n] = spec_power[n] / np.median(spec_power[n])

# -- Plot the spectragram
plt.figure()
ax = plt.subplot(111)
ax.pcolormesh(bins, freqs, np.log10(spec_power))
plt.xlabel('Time since GPS {0} (s)'.format(start))
plt.ylabel('Freq (Hz)')
plt.axis([0, stop-start, 0, fs/2])
```

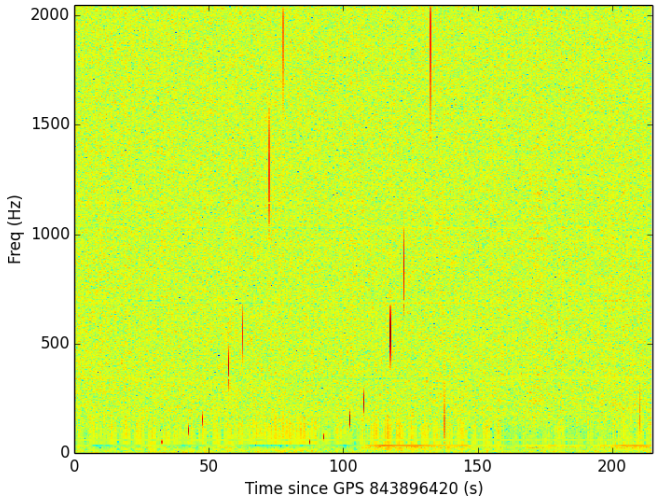
LIGO

Find a hardware injection



Find a hardware injection

- That looks like a bunch of injections. Can that be right?



- The LOSC website hosts complete injection information. Let's take a look, and see what we can find around this time

Hardware Injection Documentation

- Return to the S5 Data Release page: https://losc.ligo.org/S5/
- Near the bottom, click the link for Burst Hardware Injections
- Find the annotated log for **H1 S5 Burst Injections: HTML**
- Scroll or search the table for times included on the plot (near GPS 843896420)
  - Notice these tables can also be downloaded in machine readable formats

LIGO

Hardware Injection Logs

GPS	IFO	Waveform	Scale	Message	SNR
843896462.5	H1	wfsg100Q9	6.4	Successful	88.7
843896467.5	H1	wfsg153Q9	3.2	Successful	84.5
843896472.5	H1	wfsg235Q9	3.2	Successful	72.8
843896477.5	H1	wfsg393Q9	6.4	Successful	90.5
843896482.5	H1	wfsg554Q9	8	Successful	88.7
843896487.5	H1	wfsg850Q9	11.2	Successful	82.5
843896492.5	H1	wfsg1304Q9	12.8	Successful	61.4
843896497.5	H1	wfsg2000Q9	19.2	Successful	20.2
843896502.5	H1	wfsg3068Q9	7.2	Successful	0.0
843896507.5	H1	wfsg50Q9	96	Successful	110.3
843896512.5	H1	wfsg70Q9	32	Successful	140.3
843896517.5	H1	wfsg100Q9	12.8	Successful	177.5
843896522.5	H1	wfsg153Q9	6.4	Successful	168.9
843896527.5	H1	wfsg235Q9	6.4	Successful	145.6
843896532.5	H1	wfsg393Q9	12.8	Successful	181.0
843896537.5	H1	wfsg554Q9	16	Successful	177.4
843896542.5	H1	wfsg850Q9	22.4	Successful	165.1
843896547.5	H1	wfsg1304Q9	25.6	Successful	122.8

LIGO

Hardware Injection Logs

- The log confirms what we see in the spectrogram
  - There is a series of Burst injections
  - They are spaced in ~5 second intervals
  - There are 2 sets of injections which each rise in frequency
  - The SNRs are very loud (~100)
  - Similar information is included for all 4 data analysis working groups

LIGO

Summary

- The LOSC web site allows access to LIGO data with little or no training. Documentation includes:
  - Data access
  - Working with segments
  - Basic data analysis techniques
- LOSC data is stand-alone
  - Can be read with off-the-shelf tools
  - Includes segment and DQ information
- LOSC data is available both on the CIT cluster and on your laptop

LIGO

Everything you (or your students) need to get started is on the web

https://losc.ligo.org/

Thank you!

LIGO

Extra Slides

LOSC on the LIGO Data Grid

- LOSC data is currently available on CIT cluster
- LOSC Frame files can be accessed via ligo_data_find, exactly as other LIGO data
- Use LOSC API on LDG
  - Access LIGO data with ease
  - Move seamlessly between laptop and cluster
  - https://losc-dev.ligo.org/tutorial_gmtd/

```
$ export PYTHONPATH=$PYTHONPATH:/home/losc/py
$ ipython
> import readligo as rl
> strain, meta, dq = rl.getstrain(860000000, 860005000,'H1')
```