Introduction to LIGO data with GWpy

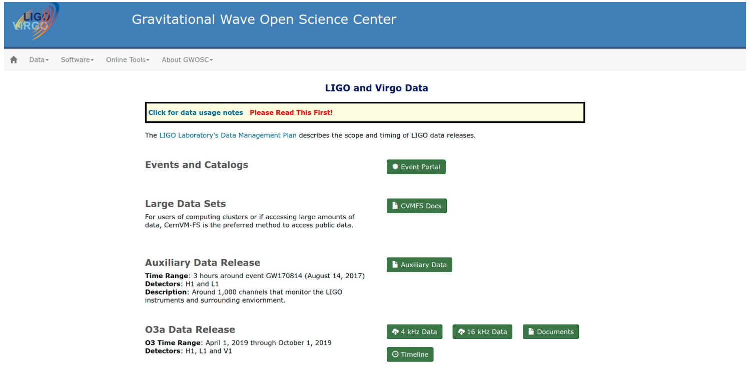
Adrian Helmling-Cornell
GWANW 2021

This Session's Goal

- Grab freely-available GW data
- Using GWpy, process the data to find the gravitational wave
- Represent the GW several different ways with GWpy plotting routines

GW Data - Getting open data

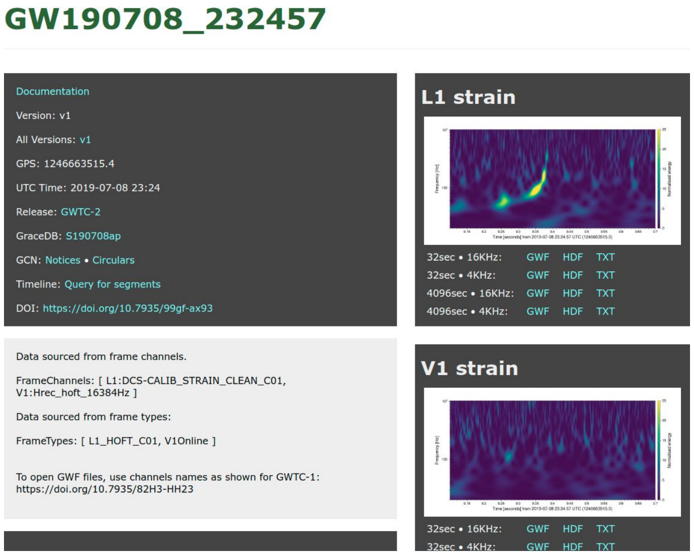
- Data from around each reported GW from O1-O3a is available for anyone to access
- How can I get LIGO data to use in my research?
- 2 options:
    - GWOSC (Gravitational Wave Open Science Center)
    - GWpy



Available data on GWOSC. https://www.gw-openscience.org/data/.

GW Data - Getting open data

- GWOSC advantages:
  - Easy to navigate UI
  - No specific programming knowledge needed to download data
- GWOSC disadvantages
  - Slow for multiple data pulls
    - O4 event rate: ~1/day
- Data available in .gwf, .hdf5, .txt formats



GWOSC data for a GW event. From [https://www.gw-opensciene.org/eventapi/html/GWTC-2/GW190708_232457/v1/](https://www.gw-openscience.org/eventapi/html/GWTC-2/GW190708_232457/v1/).

GW Data - Getting open data

- GWOSC alternative: GWpy
- GWpy is a Python package that integrates LIGO data-fetching routines with common astronomy, numerical and graphical demands
- Also works with LALSuite (LIGO Algorithm Library) code
  - C code wrapped with Python using SWIG
- More information
  - https://gwpy.github.io/docs/stable/index.html

GW Data - Getting open data

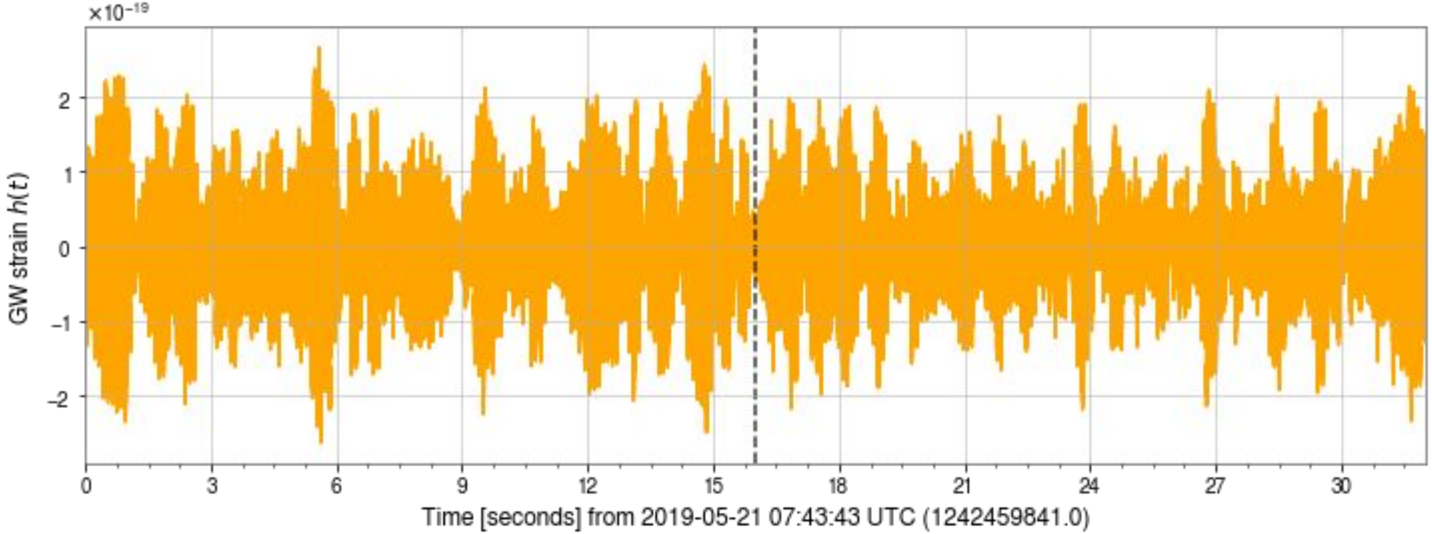
- Even if you aren't familiar with Python, it's simple to pull GW data with GWpy
- Get GWpy: for Python version 3.6+
  - Conda install -c conda-forge gwpy
  - Python -m pip install gwpy
  - Additional constraints:

Name	Constraints
astropy	>=3.0.0
dqsegdb2	
gwdatafind	
gwosc	>=0.5.3
h5py	>=2.7.0
ligo-segments	>=1.0.0
ligotimegps	>=1.2.1
matplotlib	>=3.1.0
numpy	>=1.12.0
dateutil	
scipy	>=1.2.0
tqdm	>=4.10.0

GWpy required auxiliary packages. From https://gwpy.github.io/docs/stable/install/index.html.

GW Data - Getting open data

- Using GWpy 2.0.3 for in the LIGO igwn-py36 environment
- Retrieve data for GWTC-2 event GW190521_074359



32 seconds of publicly available H1 GW strain data centered on the approximate time of GW190521_0743359.

GW Data - Getting open data

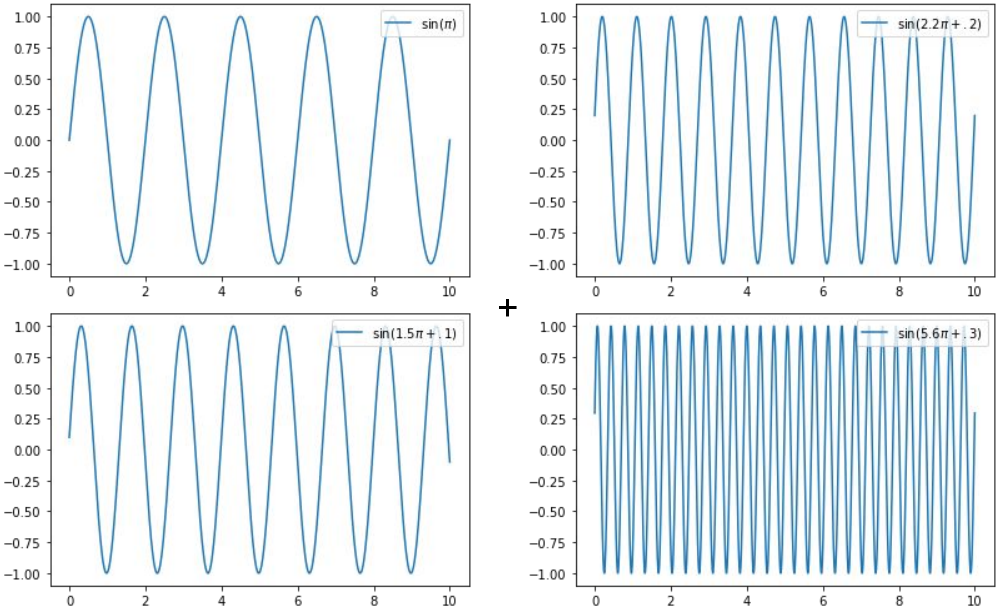
- What is produced by the fetch_open_data() method?
  - Strain data
    - DeltaL/L
  - Metadata
    - Channel name
    - Time/frequency information
    - Observatory
  - Data quality information
- For each catalog event, GWOSC/GWpy provides 4096 seconds of 16384 & 4096 Hz data centered on the event time
  - Calibration lines and known environmental noise in the data are removed
  - Not useful below 10 Hz

GW Data - Frequency domain GW data

- We're also interested in the frequency-domain content of the signal
- Reminder: a function can be rewritten as a combinations of sines and cosines
- Fourier series:

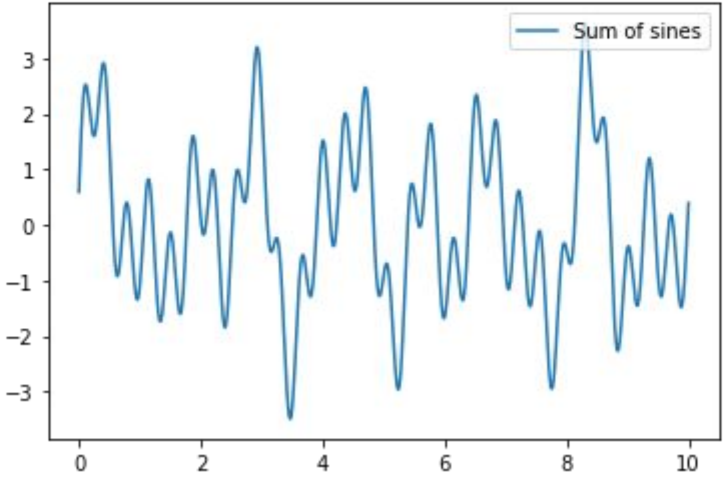
$$f(x) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi x}{L}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi x}{L}\right)$$

GW Data - Frequency domain GW data



+

=



Adding sine waves to produce some function.

GW Data - Frequency domain GW data

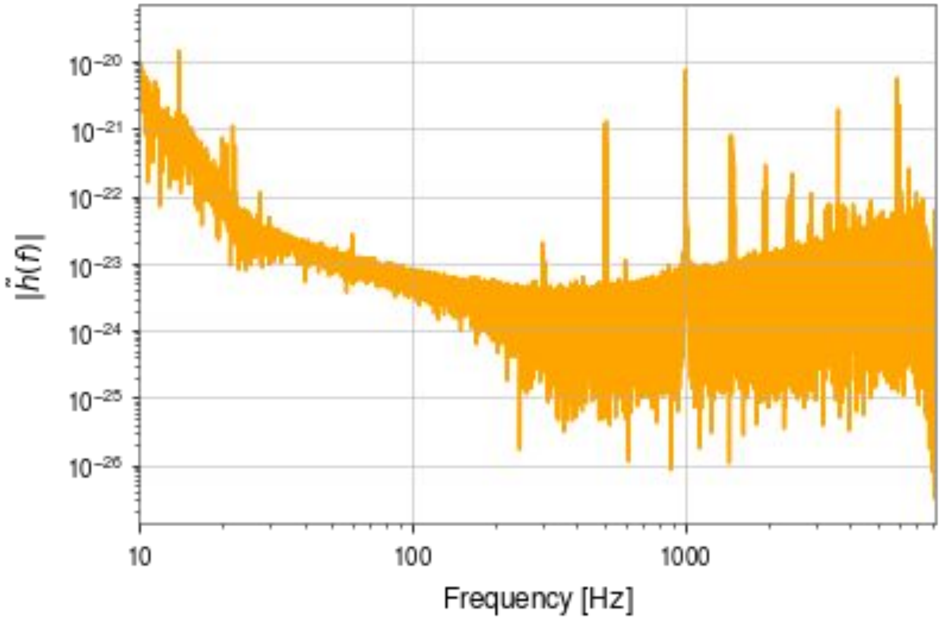
- For a given time-domain signal, we can determine both the frequencies of the sinusoids used to “construct” it as well as their amplitudes
- We can write this prescription for the time domain function as a function of frequency

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t} dt$$

$$x(t) = \int_{-\infty}^{\infty} \tilde{x}(f) e^{2\pi i f t} df$$

GW Data - Frequency domain GW data

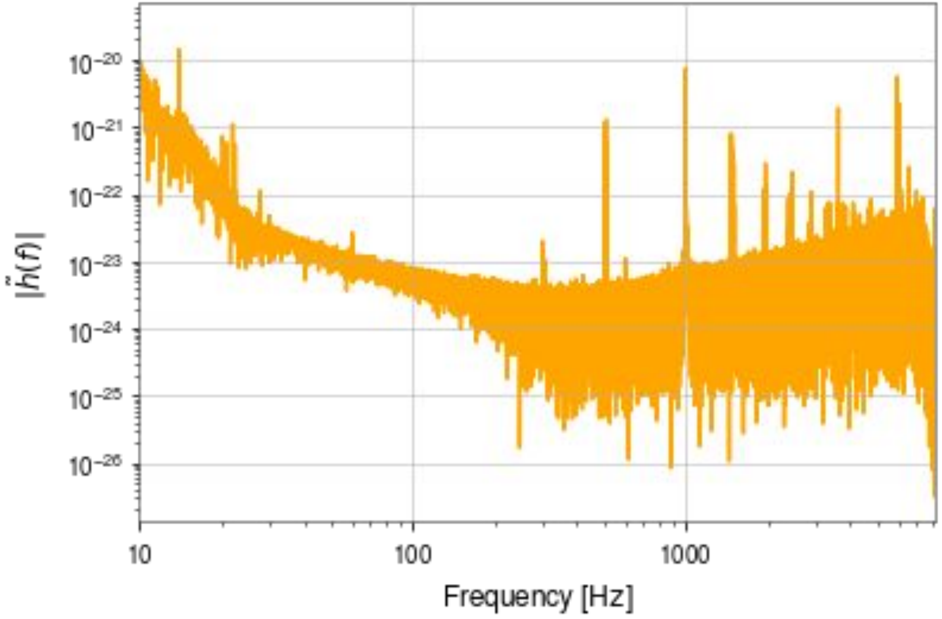
- We can take (fast) Fourier transform of the time-domain signal to see $\tilde{h}(f)$



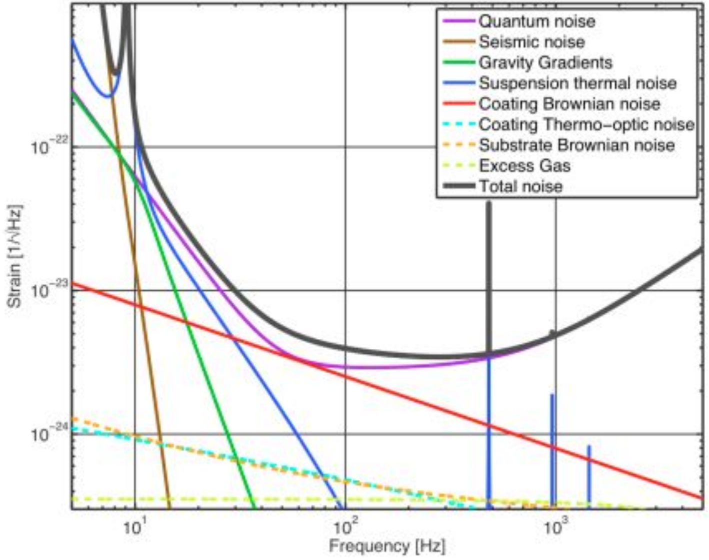
FFT of the H1 timeseries data.

GW Data - Frequency domain GW data

- We can take (fast) Fourier transform of the time-domain signal to see $\tilde{h}(f)$



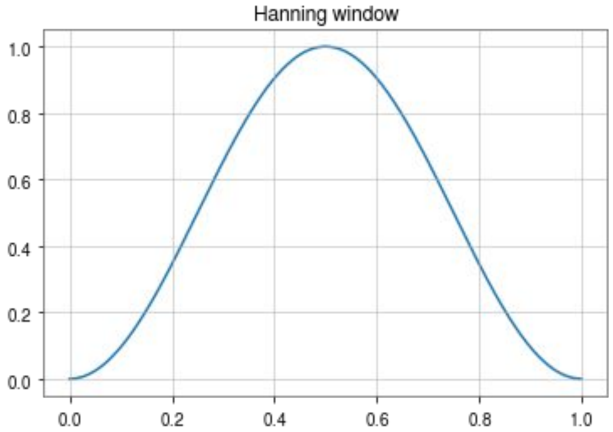
FFT of the H1 timeseries data.



Theoretical aLIGO noise budget. From LIGO DCC [T1400316-v5](T1400316-v5)

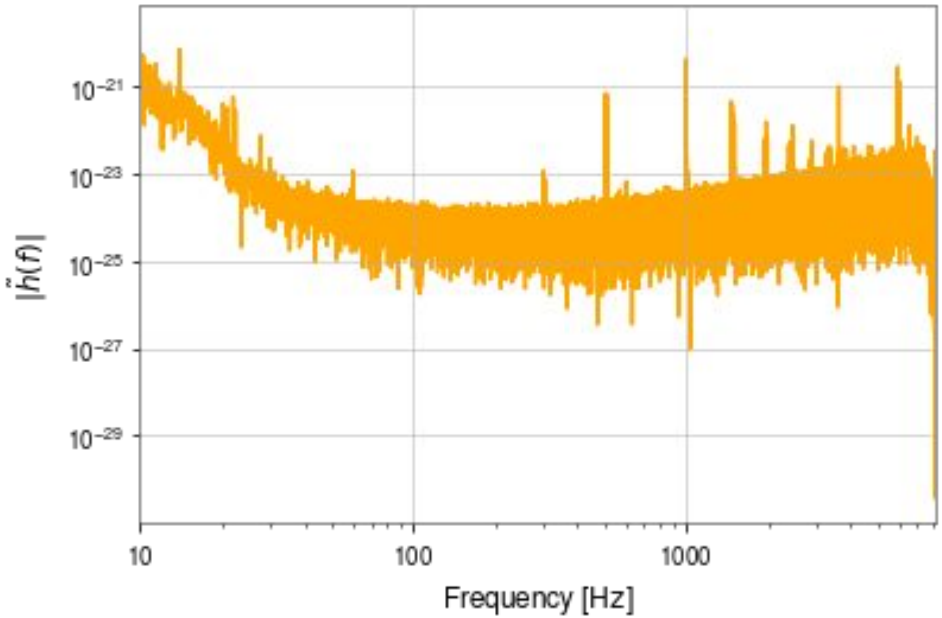
GW Data - Frequency domain GW data

- Discrepancy between our plot and LIGO sensitivity due to an underlying assumption of the FT process
  - FFTs assume the data is periodic
  - Endpoints of the signal introduce large discontinuities - “spectral leakage”
- We can limit the errors introduced by “windowing” the data
  - Combine the signal with a function that privileges information far from the endpoints of the data

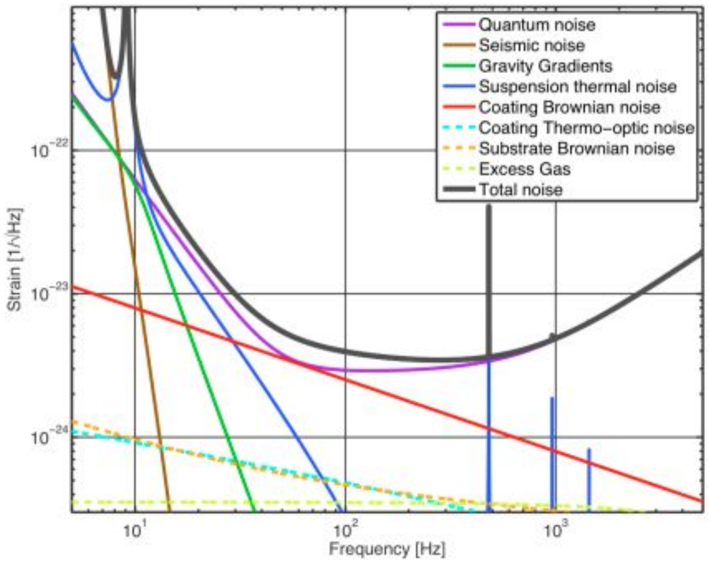


Hanning window - the window function of choice.

GW Data - Frequency domain GW data



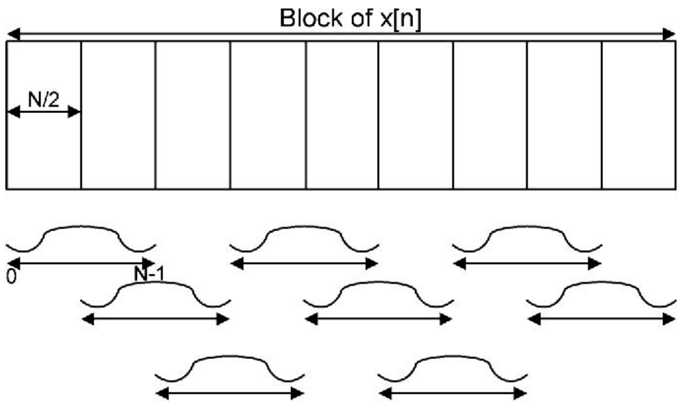
Windowed FFT of the H1 timeseries data.



Theoretical aLIGO noise budget. From LIGO DCC [T1400316-v5](T1400316-v5)

GW Data - Frequency domain GW data

- The spectrum produced by the FFT is nice, but is highly variable
    - Instrument glitches
    - Environmental noise
- Preferable to average several shorter FFTs together to make spectra
- Welch's method
    - Split the signal into overlapping chunks
    - Multiply each chunk's signal by a window function
    - Perform an FFT on each chunk
    - Average the amplitude reported in each frequency bin across each FFT



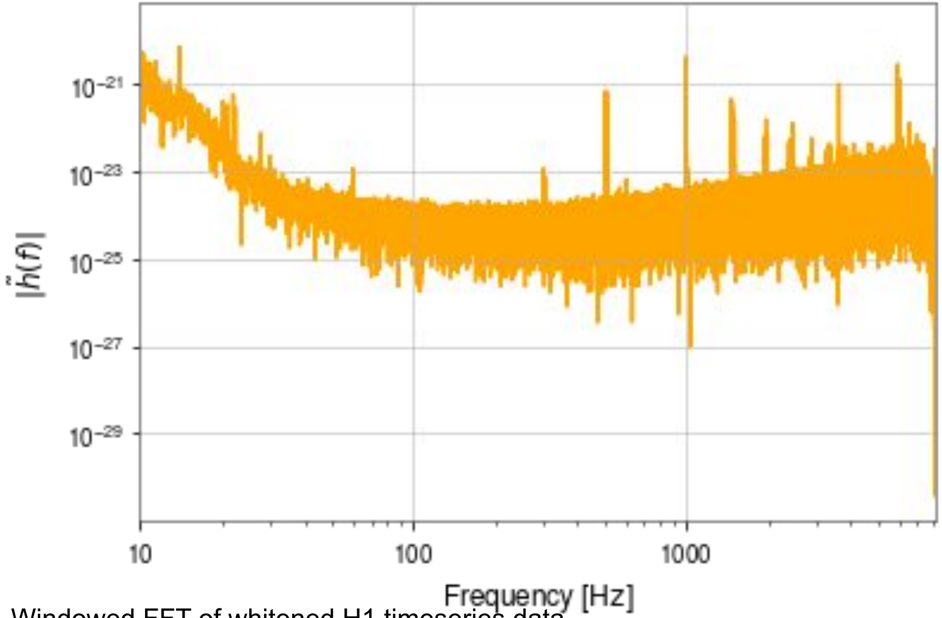
Schematic of the chunking & windowing process of Welch's method. From https://doi.org/10.1109/TCSI.2013.2264711.

GW Data - Frequency domain GW data

- Result - amplitude spectral density of the GW data
- Signal amplitude in frequency bin per sqrt(Hz)
- Square root of the power spectral density
    - Power in frequency bin per Hz

GW Data - Frequency domain GW data

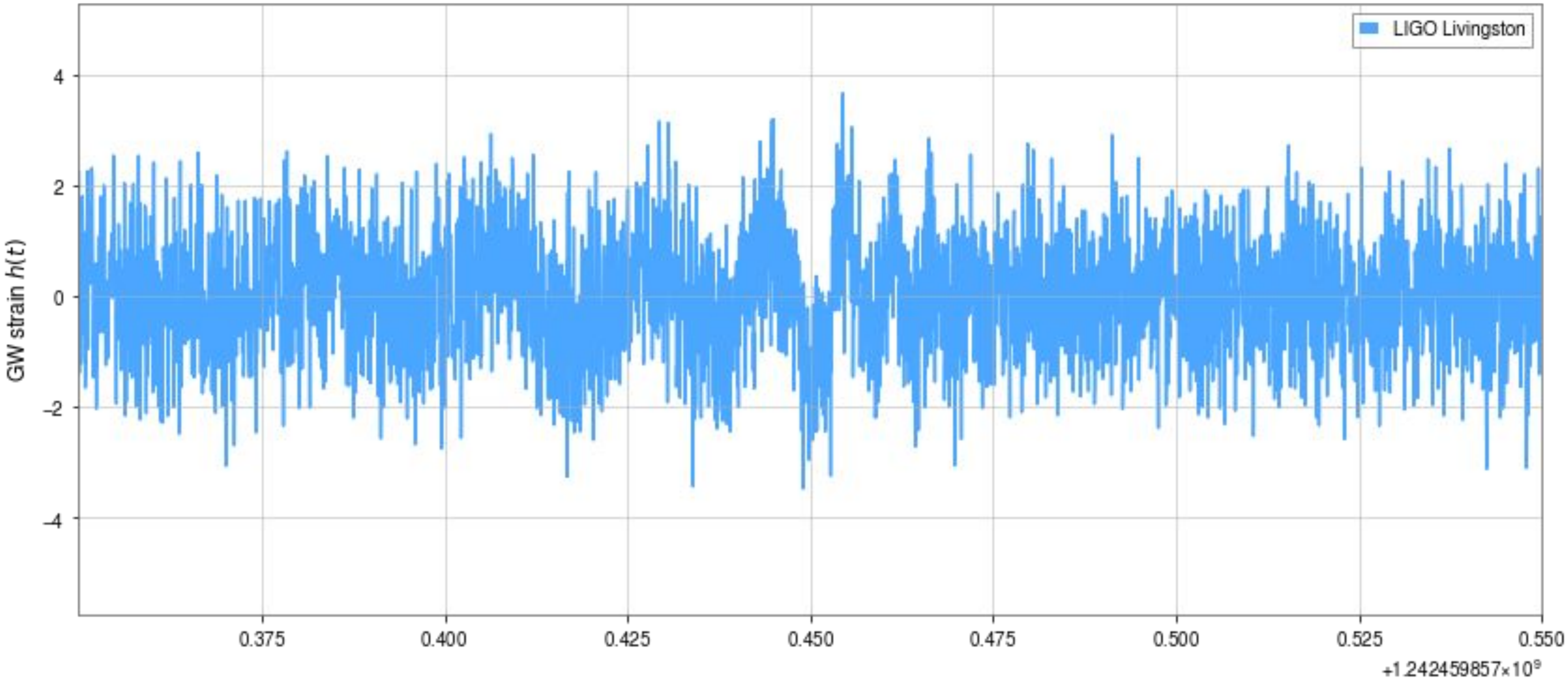
- Why couldn't we see the GW in the timeseries beforehand?
  - Compare amplitude at low frequencies to amplitude at “detection-band” frequencies
- We can use GWpy to suppress the response at low frequencies via whitening



Windowed FFT of whitened H1 timeseries data.

GW Data - Frequency domain GW data

- Clear inspiral present in Livingston data



GW Data - Spectrograms, PSDs and Q-transforms

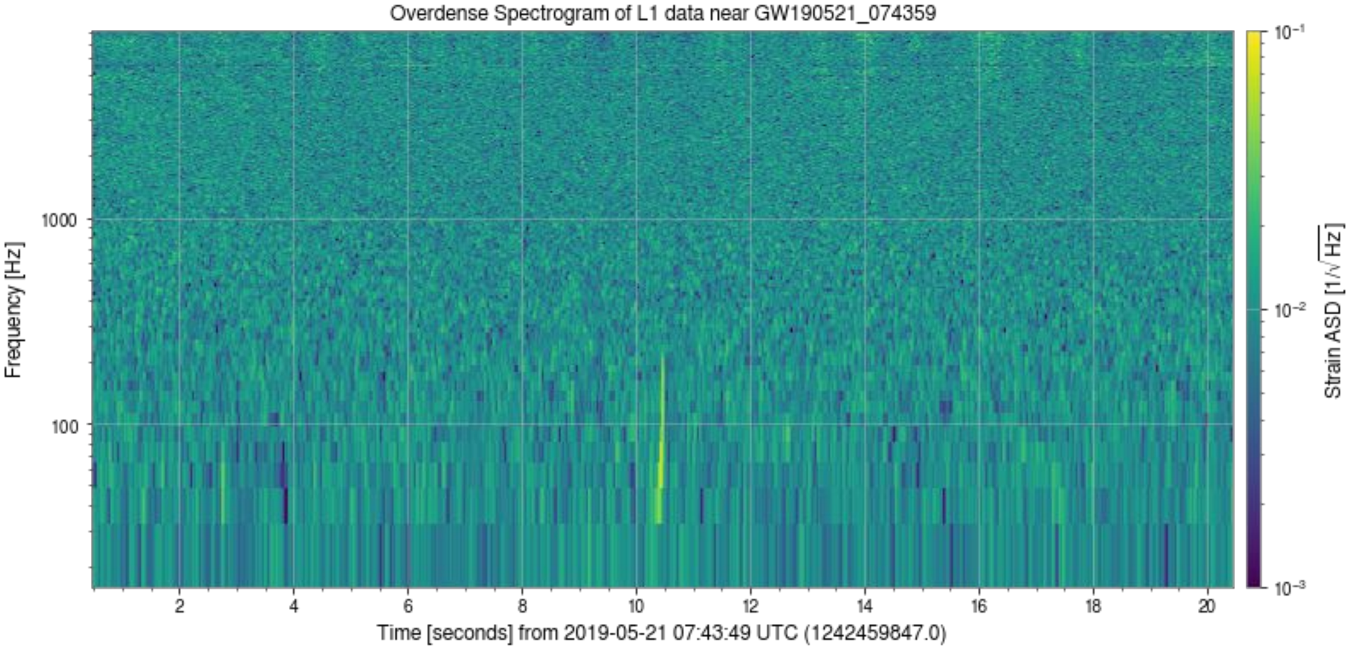
- We can use ASDs to produce other powerful visualization tools
- A spectrogram combines a series of ASDs on a chunk of data
    - Shows frequency-domain evolution over time
    - Time-frequency-amplitude plots
- At each time, the spectrogram plots the amplitude in a certain frequency bin
    - Changing signal characteristics appear as changing colors on the spectrogram

GW Data - Spectrograms, PSDs and Q-transforms

- GWpy provides two methods or computing spectrograms
  - spectrogram() shows an averaged ASD for each time slice
  - spectrogram2() computes only FFT per time slice
- spectrogram() is preferable for long periods of data
- spectrogram2() sensitive to short noise bursts - no averaging to smooth each time slice

GW Data - Spectrograms, PSDs and Q-transforms

- These provide us another way to see a GW signal
- Highly overlapping FFTs used to make more features in the data apparent



GW Data - Spectrograms, PSDs and Q-transforms

- Our code for the spectrogram has a square root in it
    - GWpy makes spectrograms with *power* spectral densities rather than amplitude spectral densities
- When we calculated our GW ASD, we found ~ $|\tilde{h}(f)|$
- $\dfrac{|\tilde{h}(f)|^2}{T}$ is the energy spectral density divided by the time, or the PSD
- Units: power in frequency bin per Hz

GW Data - Spectrograms, PSDs and Q-transforms

- Q-transform similar to an FFT
  - Analysis window inversely proportional to frequency

$$x(\tau, f) = \int_{-\infty}^{\infty} \tilde{x}(\phi + f)\tilde{w}^*(\phi, f)e^{2\pi i\phi\tau}d\phi$$

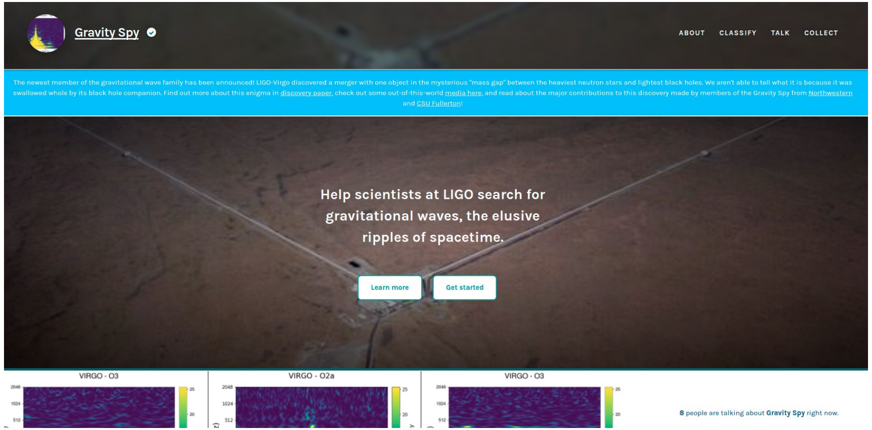
- Q-transform is the IFFT of some FFTed data plus a frequency shift times a windowing function
- Useful for describing evolution of time-frequency features that happens over short timescales (like a GW)

GW Data - Spectrograms, PSDs and Q-transforms

- The “Q” in Q-transform comes from the quality factor

$$Q = \frac{f_0}{\Delta f}$$

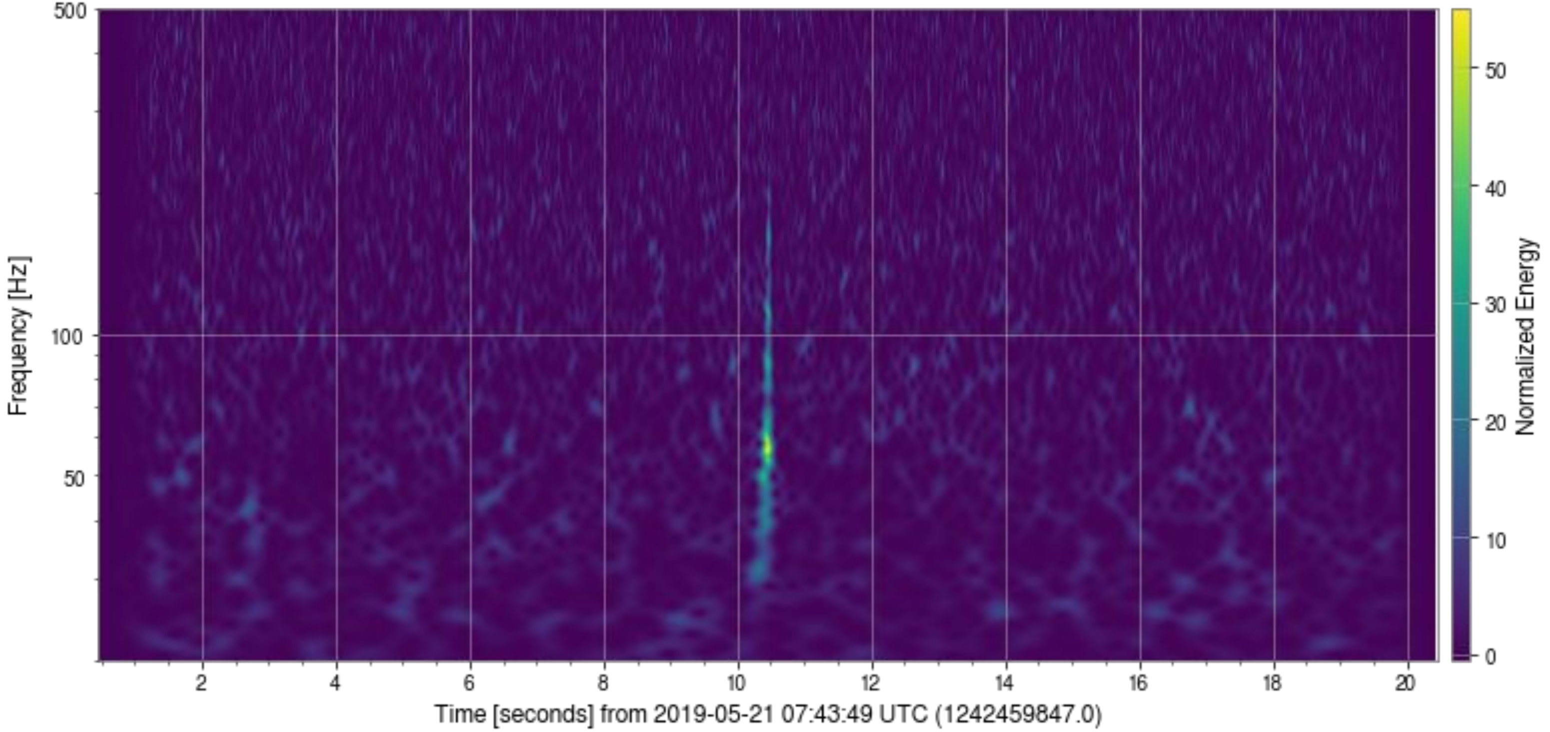
- Logarithmically spaced frequency bins - closer to how humans perceive frequency/amplitude differences



Gravity Spy Zooniverse page. https://www.zooniverse.org/projects/zooniverse/gravity-spy.

GW Data - Spectrograms, PSDs and Q-transforms

- Livingston GW signal



- Some things we can work on:
    - Installing & configuring GWpy
    - Comparing GW events and noise features between interferometers and epochs
    - Reviewing this information by going through the GWOSC tutorials