GW data analysis: signal processing

18.1.21, 2.2.21

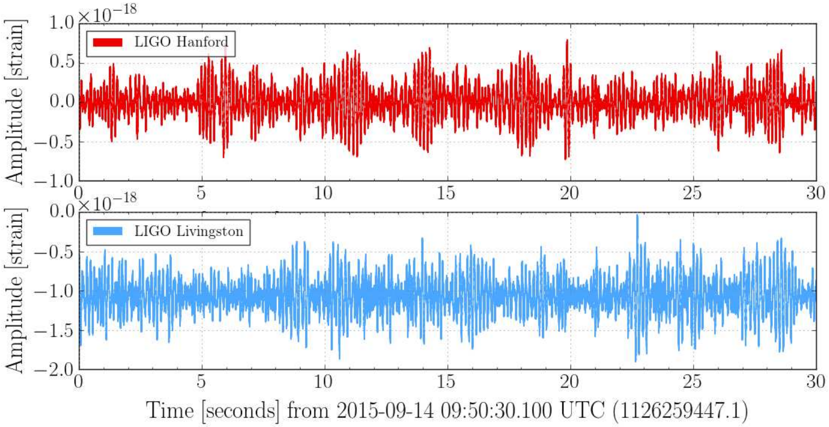
General schedule

- ⋆ History
- ⋆ Introduction to general relativity
- ⋆ Detection principles
- ⋆ Detectors
- ⋆ Binary black-hole system
- ⋆ Bursts and continuous waves
- ⋆ Rates and populations & cosmology
- ⋆ Stochastic GW background. Tests of general relativity using GWs
- ⋆ Data analysis: signal processing
  - ⋆ Time and frequency domain
  - ⋆ Useful signal processing methods
  - ⋆ Signal detection
- ⋆ Data analysis: parameter estimation

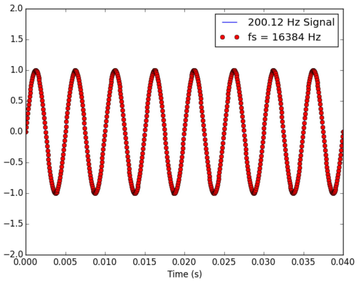
GW data interferometer output

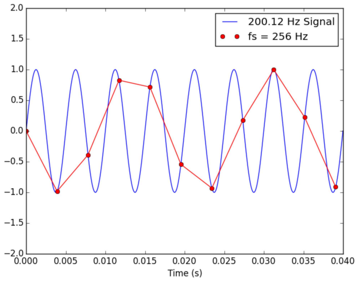
- ⋆ LIGO/Virgo raw data is uniformly-sampled time series of calibrated GW strain $h(t) = \Delta L(t)/L$
- ⋆ Sampling at a sampling frequency $f_s = 16384$ Hz for LIGO, $f_s = 20000$ Hz for Virgo,
- ⋆ In addition, $\simeq$200000 auxiliary channels to monitor conditions of the detectors,
- ⋆ $\simeq$50 MB of data per hour per detector.

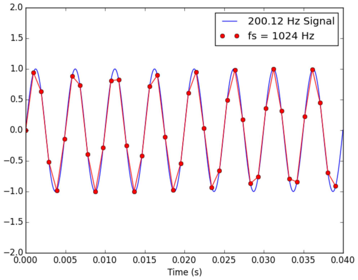
Raw GW $h(t)$ data time series



Sampling theorem. Nyquist frequency

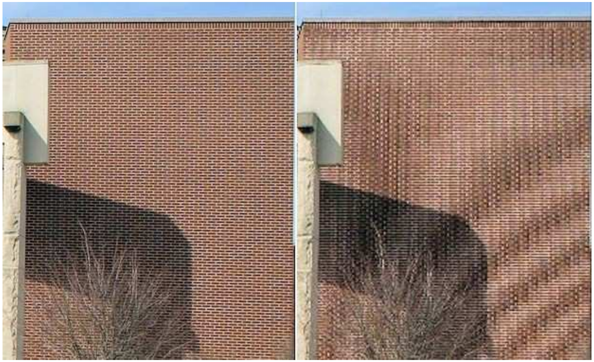






- ⋆ Nyquist frequency $f_N = f_s/2$,
- ⋆ Data can only accurately represent frequency content below the Nyquist frequency,
- ⋆ Higher frequency signals will be lost (or "aliased" to lower frequencies)

Sampling theorem. Nyquist frequency



this effect is called **aliasing**

examples abound in digital audio, imaging, and film

The Fourier Transform

Any function can be represented as a sum of sines and cosines:

$$h(t) = \sum_{n=0}^{\infty} A_n \cos(\frac{n\pi t}{N}) + \sum_{n=0}^{\infty} B_n \sin(\frac{n\pi t}{N})$$

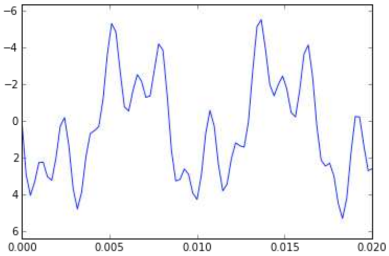
When we transform our function or time (or space) into the "frequency domain", we are **projecting f(x) onto an orthogonal basis of sines and cosines**.

Fourier transform
$$\widetilde{x}(f) = \int_{-\infty}^{\infty} dt\, x(t) e^{-i2\pi ft}$$

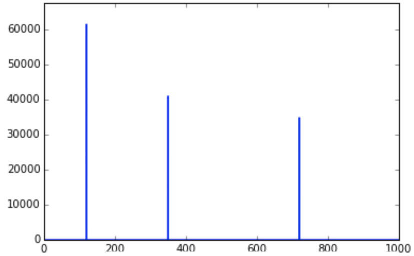
Inverse Fourier transform
$$x(t) = \int_{-\infty}^{\infty} df\, \widetilde{x}(f) e^{i2\pi ft}$$

Another way to think about it: when we take a Fourier transform **we are decomposing the function into its component frequencies**.

Time domain → frequency domain



Time (s)



Frequency (Hz)

h(t) – Position as a function of time

H(f) – Amplitude as a function of frequency

h(t) = 3 * sin(2*pi*120*t) +
2 * sin(2*pi*350*t) +
1.5* sin(2*pi*720*t)

|H(120 Hz)| = 3
|H(350 Hz)| = 2
|H(720 Hz)| = 1.5
H(f) = 0 otherwise

⟵ Fourier Transform ⟶

Slide from J. Kanner

Filtering specific frequencies and times: convolution

For two signals h and g, a convolution $(h \star g)$ is an operation that consists of sliding signals against each other and integrating:

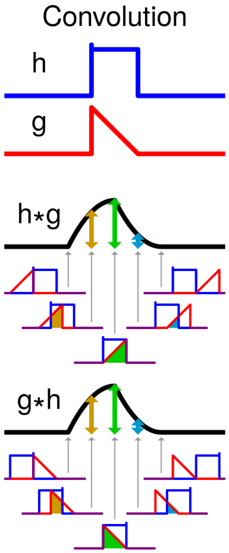
$$(h \star g)(t) = \int_{-\infty}^{\infty} h(\tau) g(t-\tau) d\tau.$$

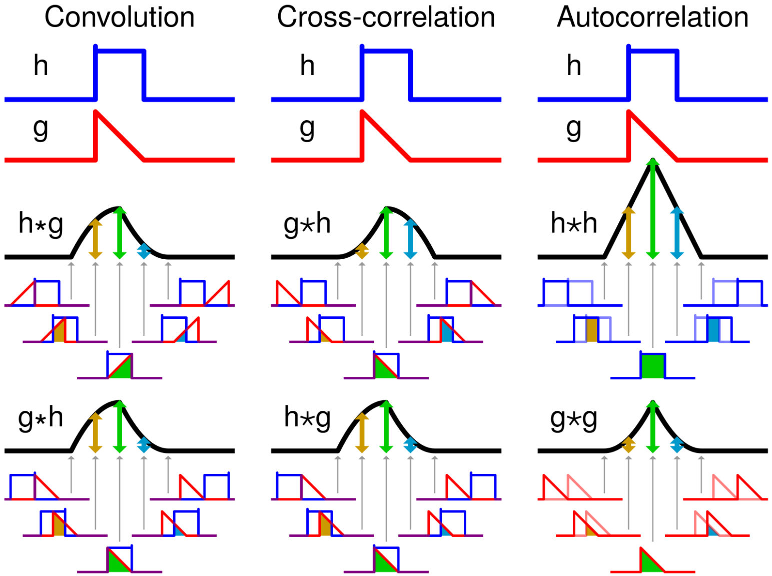
In frequency domain (after applying the Fourier transform) convolution becomes simple multiplication:

$$\widetilde{(h \star g)}(f) = \tilde{h}(f)\tilde{g}(f).$$

Symmetry of h is the reason $g \star h$ and $h \star g$ are identical in this example

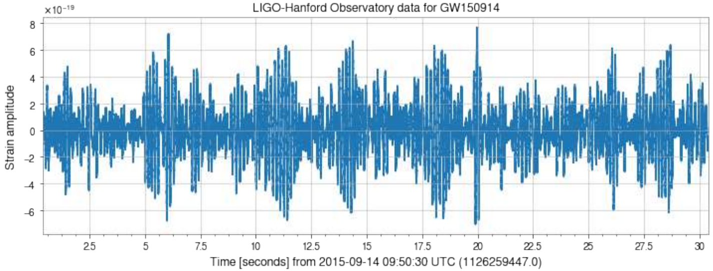
(`en.wikipedia.org/wiki/Convolution`)




Convolution
Cross-correlation
Autocorrelation
h
g
h
g
h
g
h∗g
g⋆h
h⋆h
g∗h
h⋆g
g⋆g

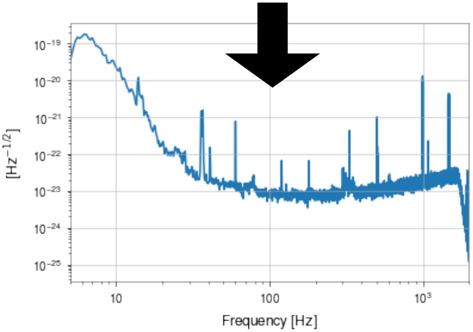
Time domain $\rightarrow$ frequency domain

Time series (strain vs time)



Amplitude spectral density

($\mathrm{Hz}^{-\frac{1}{2}}$ vs. frequency)



Power Spectral Density (PSD)

Parseval's theorem:

$$\int_{-\infty}^{\infty} dt\, |x(t)|^2 = \int_{-\infty}^{\infty} df\, |\tilde{x}(f)|^2$$

$\Rightarrow$ Total energy in the data can be calculated in either time domain or frequency domain

Units:

$|\tilde{x}(f)|^2$ Energy spectral density
(normalize by 1/T to get power)
Signal energy per unit frequency (per Hz)

$|\tilde{x}(f)|$ $\propto$ Amplitude spectral density
(sqrt of power for each discrete frequency)
Signal amplitude per unit frequency (per *sqrt Hz*)

Estimating the PSD

Step 0: Take a **Fast Fourier Transform (FFT)**, which is any algorithm useful for quickly estimating the **Discrete Fourier Transform** that describes a **discrete time series**.

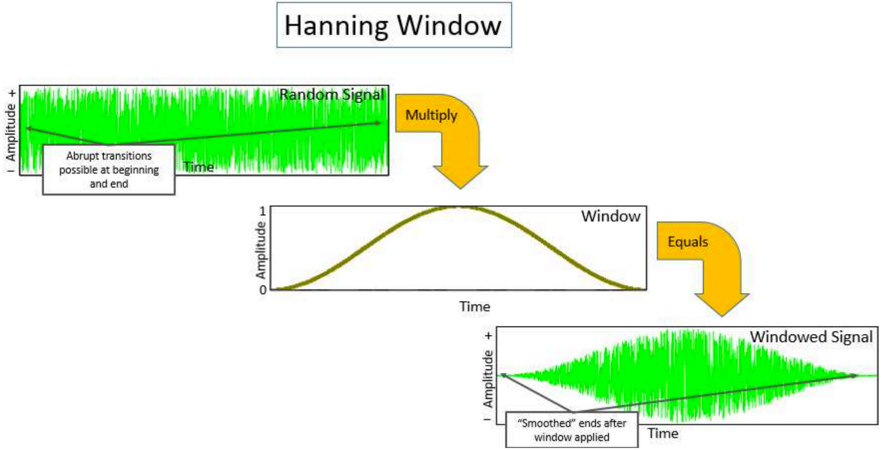
$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} \\ &= \sum_{n=0}^{N-1} x_n \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)], \end{aligned}$$

Need to shift our thinking to discretized data; **frequency bins instead of continuous smooth sinusoids**

(The FFT reduces the computational cost from $O(N^2)$ to $O(N \log N)$ for a transform of length N).

Estimating the PSD

Step 1: Apply a window to your data (if it's linear! as a time series is) to prevent **spectral leakage** from the assumption the signal is periodic.



see more at `https://en.wikipedia.org/wiki/Window_function`

Estimating the PSD

A single windowed FFT is unbiased (i.e. will give the correct mean PSD), but has **high variance**.

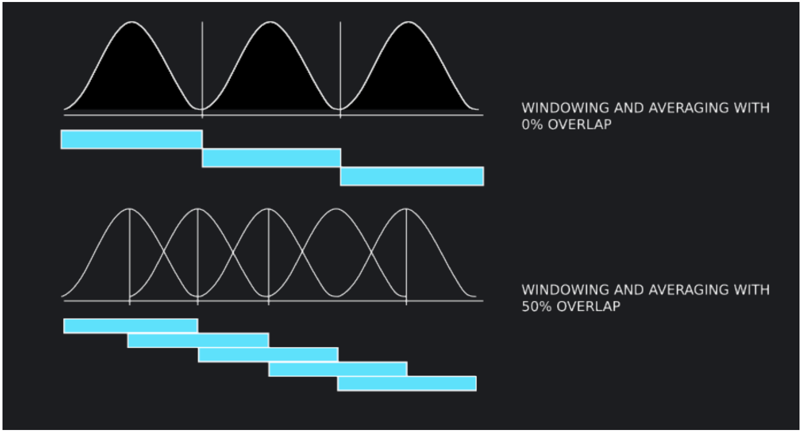
Solution: average several FFTs!

Step 2: Divide your data into shorter time segments; take a windowed FFT of each, and average these together.

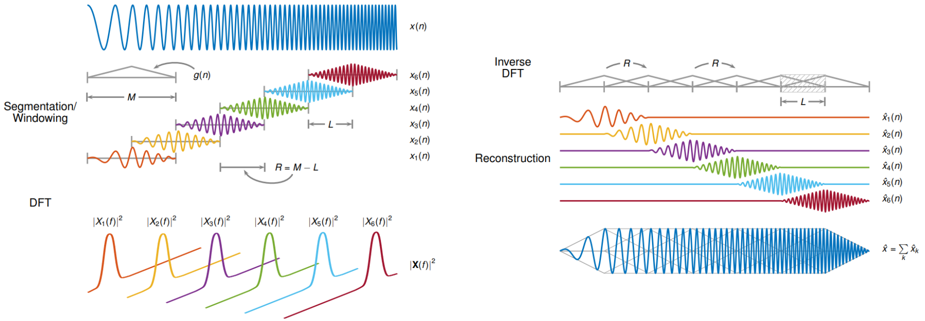
Note you lose some frequency resolution this way.

Welch's method averages the mean value for each frequency bin across FFTs, with some overlap in the data analyzed.

Estimating the PSD. Averaging FFTs


WINDOWING AND AVERAGING WITH 0% OVERLAP
WINDOWING AND AVERAGING WITH 50% OVERLAP

Reconstructing time domain from Short-Time FTs



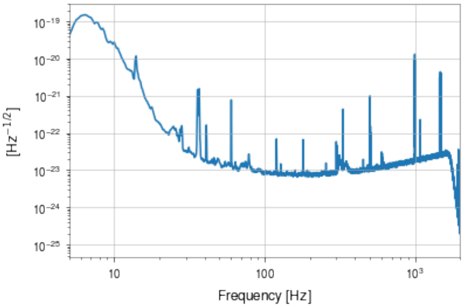
`https://www.mathworks.com/help/signal/ref/istft.html`,

”Invertivility of overlap-add processing”:

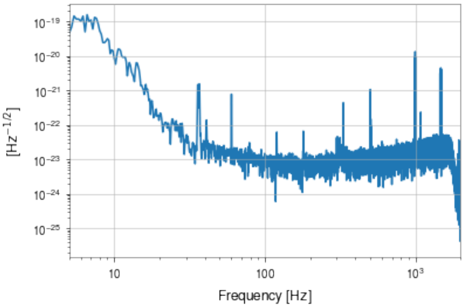
`https://gauss256.github.io/blog/cola.html`

Example: averaging FFTs

FFT length = 5 seconds
Overlap = 2 seconds
120 averages

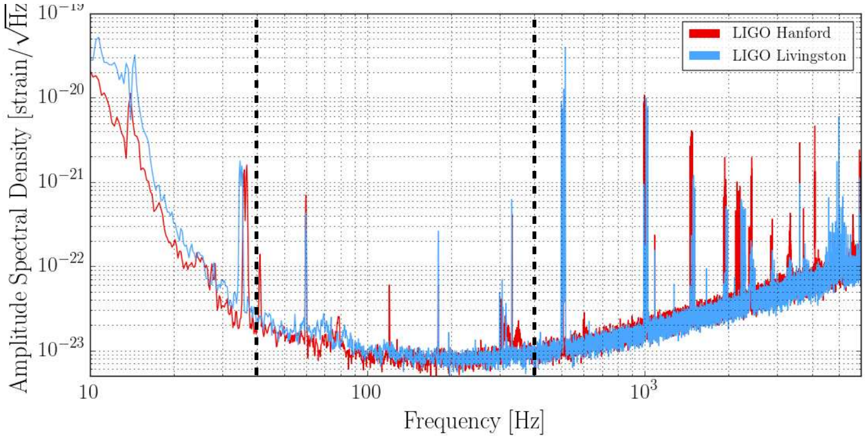


FFT length = 5 seconds
Overlap = 2 seconds
4 averages



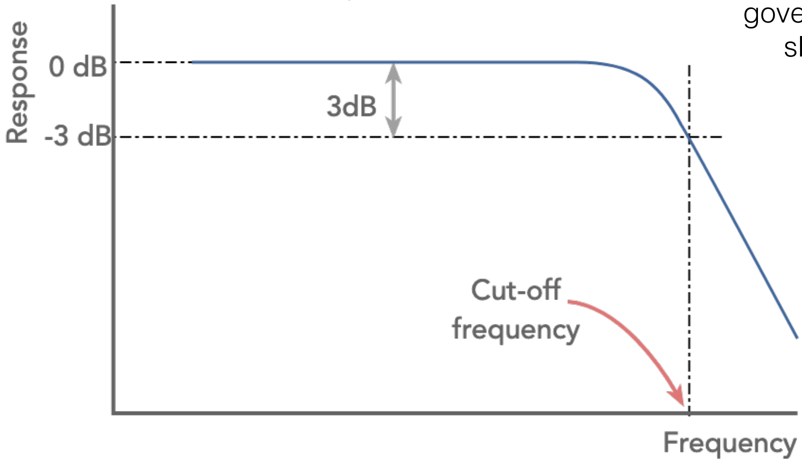
Common signal processing methods/filters

- ⋆ windows (`https://en.wikipedia.org/wiki/Window_function`)
- ⋆ low- and high-pass filters,
- ⋆ bandpass,
- ⋆ notch,
- ⋆ whitening.



Low-pass and high-pass filters

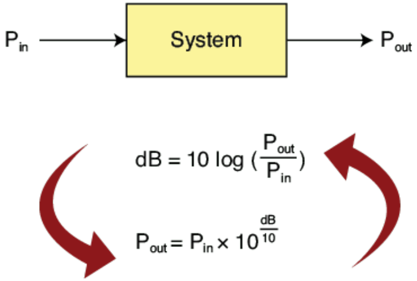
A “low pass” filter

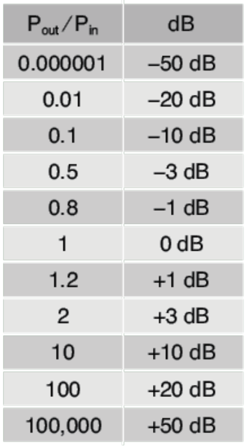


Filter “order” governs the slope

Decibel

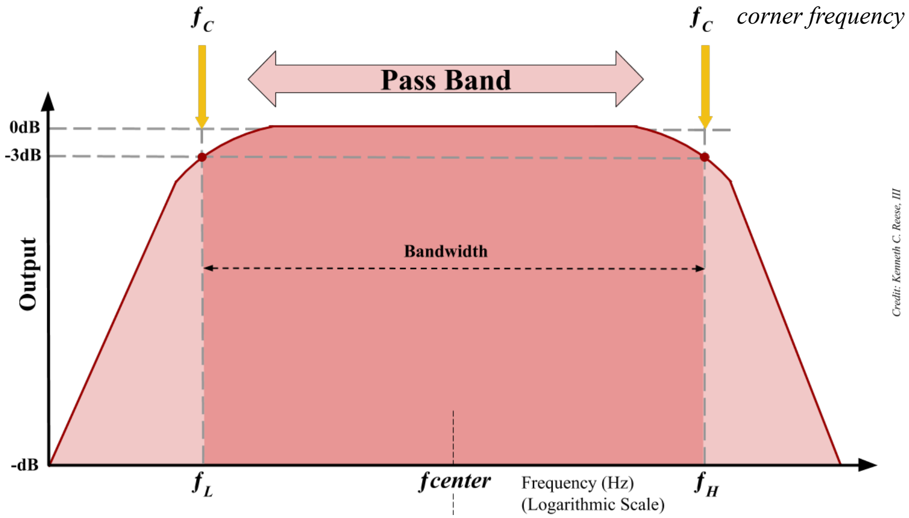
Decibel (dB) is a relative unit of measurement (one tenth of a bel, B). Expresses the ratio of one value of a power or root-power quantity to another, on a logarithmic scale (analog of astronomical magnitude: $m_1 - m_2 = -2.5 \log_{10}(I_1/I_2)$, where I_i are e.g. intensities in $[W/m^2]$)


P_{in}
System
P_{out}
$dB = 10 \log \left(\frac{P_{out}}{P_{in}}\right)$
$P_{out} = P_{in} \times 10^{\frac{dB}{10}}$



P_{out}/P_{in}	dB
0.000001	–50 dB
0.01	–20 dB
0.1	–10 dB
0.5	–3 dB
0.8	–1 dB
1	0 dB
1.2	+1 dB
2	+3 dB
10	+10 dB
100	+20 dB
100,000	+50 dB

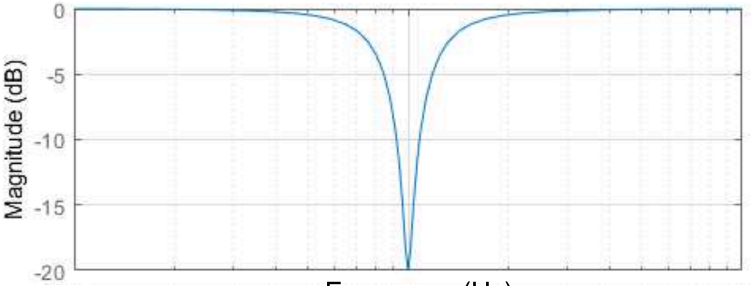
Bandpass filter



Notch filter

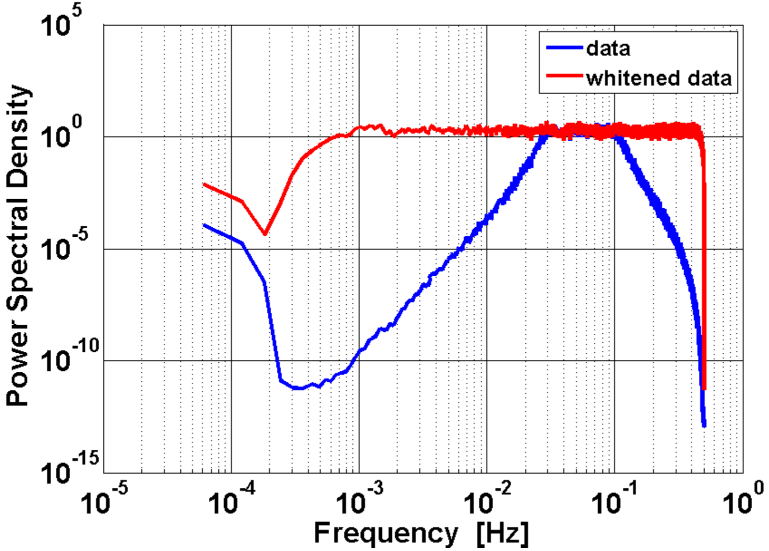
Not quite the inverse of the bandpass filter

Only described by one frequency (and the filter order)



Frequency (Hz)

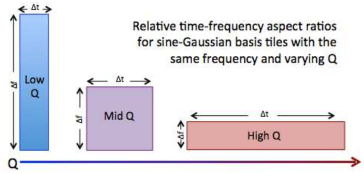
Whitening



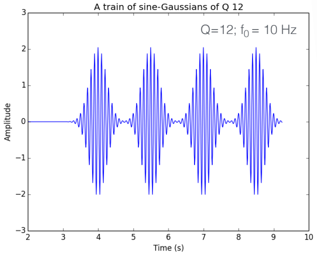
data
whitened data
Power Spectral Density
Frequency [Hz]
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}

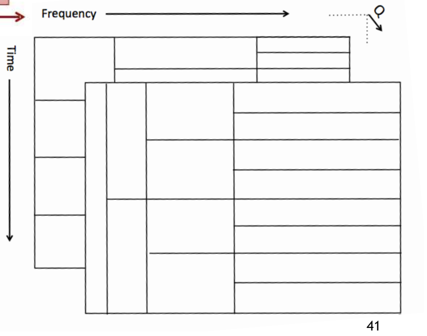
Q-transform

S. Chatterji et al. CQG (2010)
Images: McIver


Δt
Δf
Low Q
Mid Q
High Q
Q
Relative time-frequency aspect ratios for sine-Gaussian basis tiles with the same frequency and varying Q

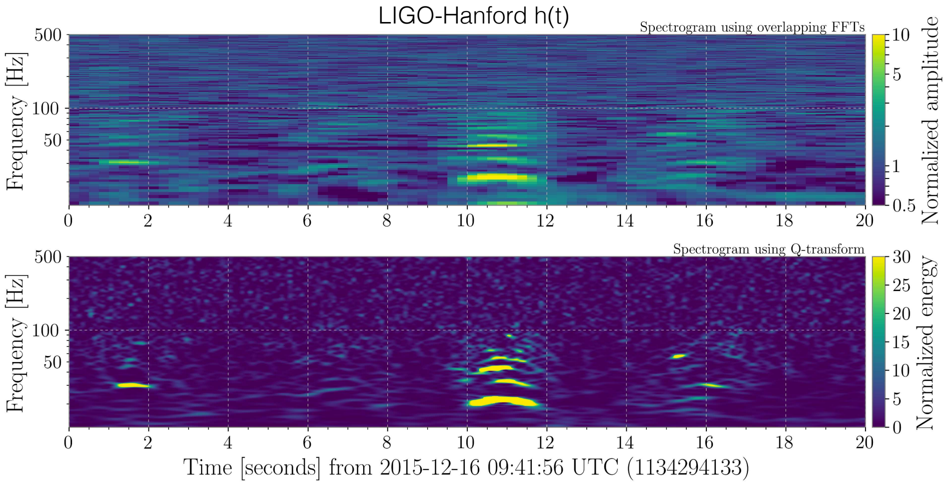
$$Q = \frac{f_0}{\Delta f}$$





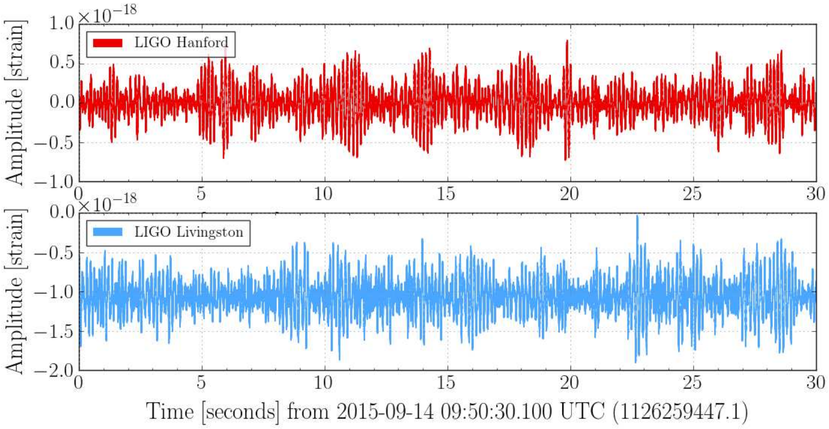
A series of filters f_k, logarithmically spaced in frequency, with the k-th filter a spectral width of Δf_k.

Time-frequency spectrogram, Q-transform



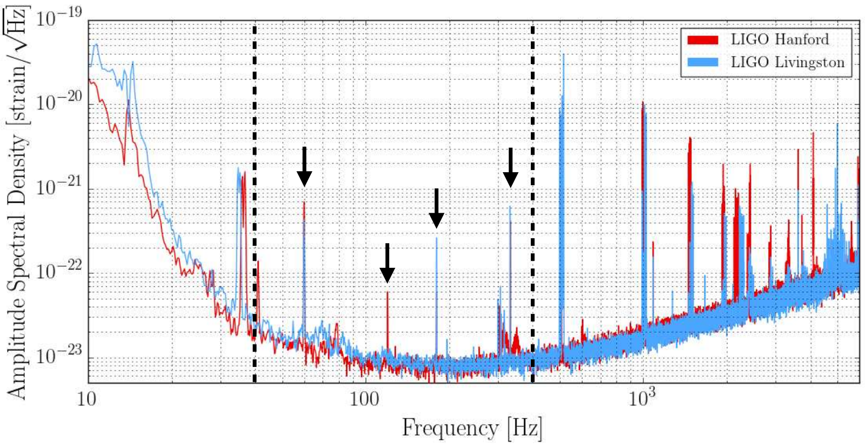
see more at `https://gwpy.github.io/docs/stable/examples/timeseries/qscan.html`

Detecting GW150914: raw $h(t)$



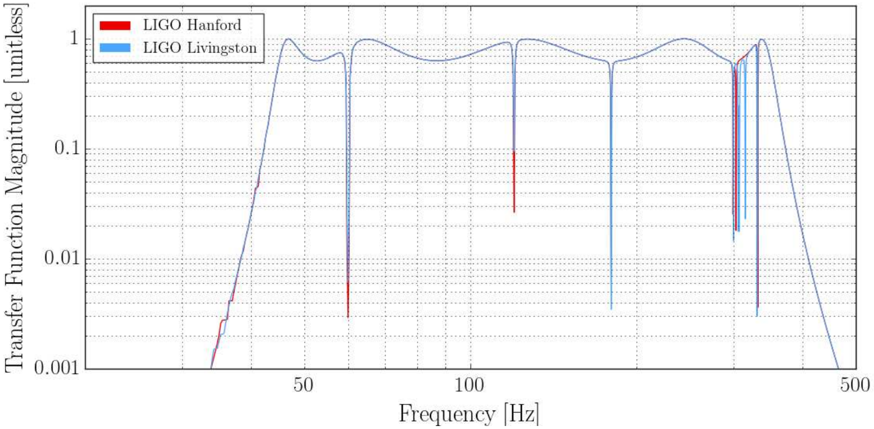
Raw data is dominated by the low-frequency content (mostly anthropogenic and seismic origin).

Detecting GW150914: ASD



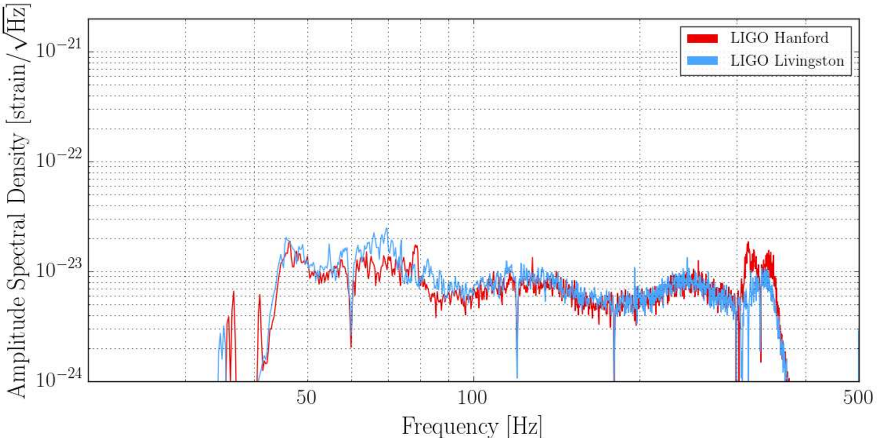
GW150914 signal is an inspiral and merger of two massive BH $\rightarrow$ restricted frequency range $\rightarrow$ bandpass filter (range indicated by vertical dashed lines). Additionally, applying the notch filter(s) to remove constant-frequency lines (calibration etc.), indicated by arrows.

Detecting GW150914: applying a transfer function

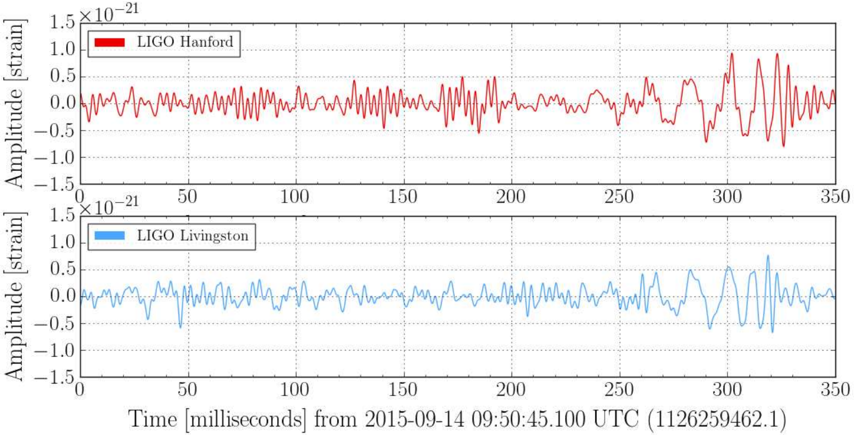


Removing specific frequencies using a sum of notch filters (transfer function).

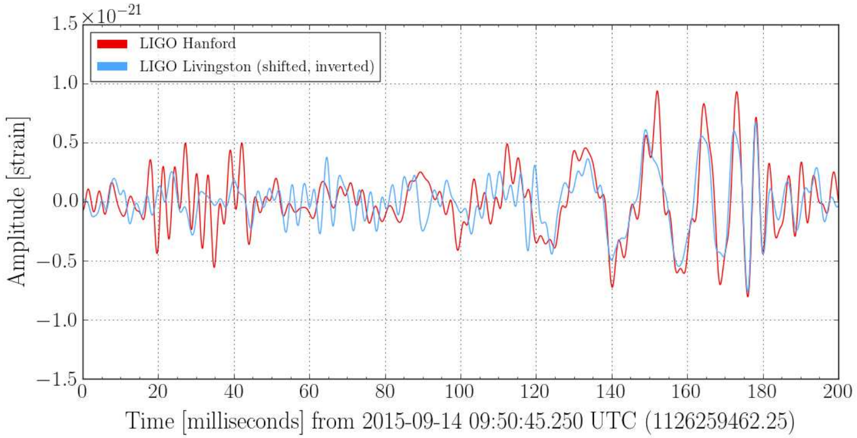
Detecting GW150914: bandpassed and notched data



Detecting GW150914: bandpassed, notched and whitened time domain data



Detecting GW150914: signal coherence

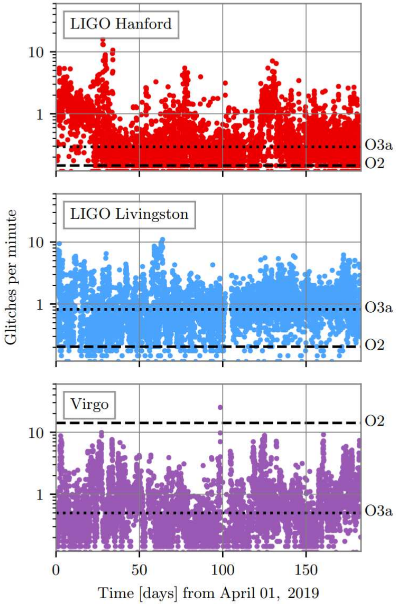


Comparison of two detectors' data: there is a coherent signal in both detector streams at the same time.

Real-life data quality

In practice, LIGO-Virgo data differs from what is expected from an idealized interferometer (stationary, Gaussian noise):

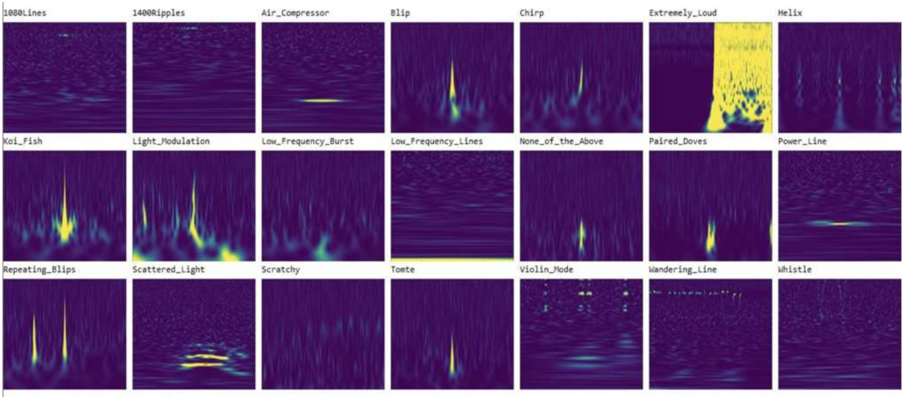
- ⋆ sensitivity curves contain ”lines” and broadband features (almost stationary),
- ⋆ temporary features: changes in various times (hours, seconds),
- ⋆ transient $f - t$ phenomena: ”glitches”, many of them have unknown source.



(GWTC-2 `arXiv:2010.14527`)

”Glitches zoo”: transient instrumental noise

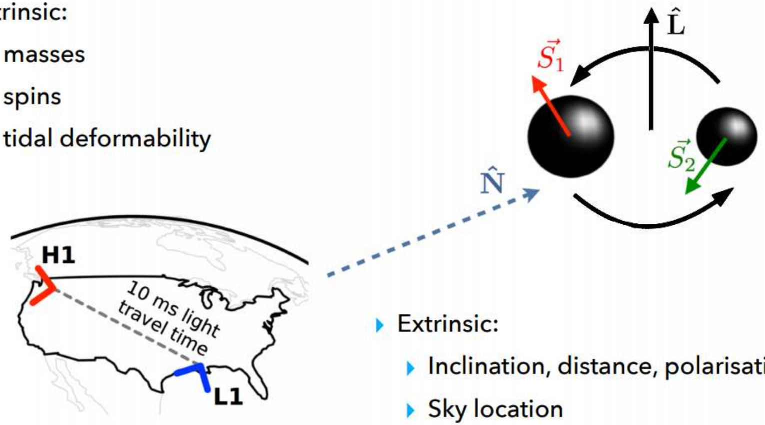
Excess power (glitches) represented as spectrograms - time-frequency maps - suitable for human-eye inspection:



- ⋆ Main problem for the sensitivity of transient searches,
- ⋆ Citizen science: `Gravity Spy`, `Reinforce` (preparation of training data for machine learning).

Binary system waveform: 15+ parameters

- Intrinsic:
  - masses
  - spins
  - tidal deformability



Credit: LIGO/Virgo

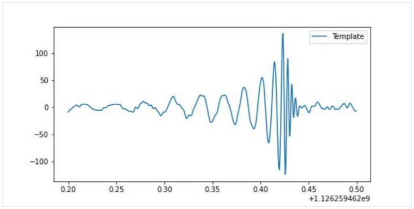
- Extrinsic:
  - Inclination, distance, polarisation
  - Sky location
  - Time, reference phase

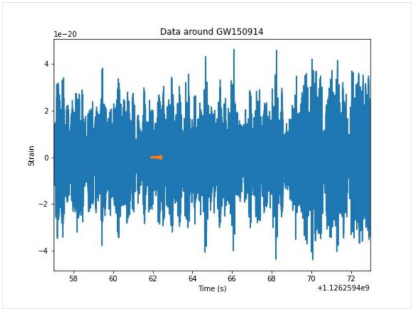
Matched filtering

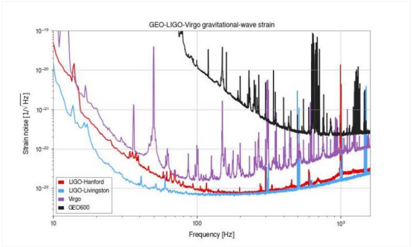
- ⋆ With the data $s(t) = n(t) + h(t)$,
- ⋆ signal template $h_t(t)$ and
- ⋆ one-sided amplitude spectral density of the noise $S_n(f)$, defined as $\langle \tilde{n}(f)\tilde{n}^\star(f')\rangle = \frac{1}{2}S_n(|f|)\delta(f-f')$,

the matched filter is an inner product

$$(s|h) = 2\int_{-\infty}^{\infty} \frac{\tilde{s}(f)\tilde{h}_t^\star(f)}{S_n(f)}\,df$$







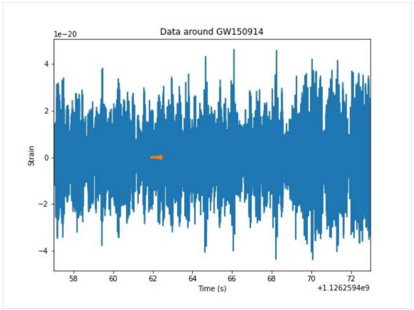
Allen et al. Phys. Rev. D 85 122006 (2012)

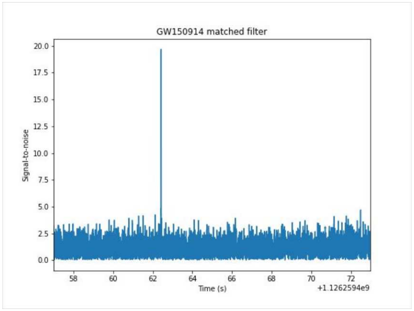
Matched filtering

The matched filter output

$$(s|h) = 2\int_{-\infty}^{\infty} \frac{\tilde{s}(f)\tilde{h}_t^\star(f)}{S_n(f)}\,df$$

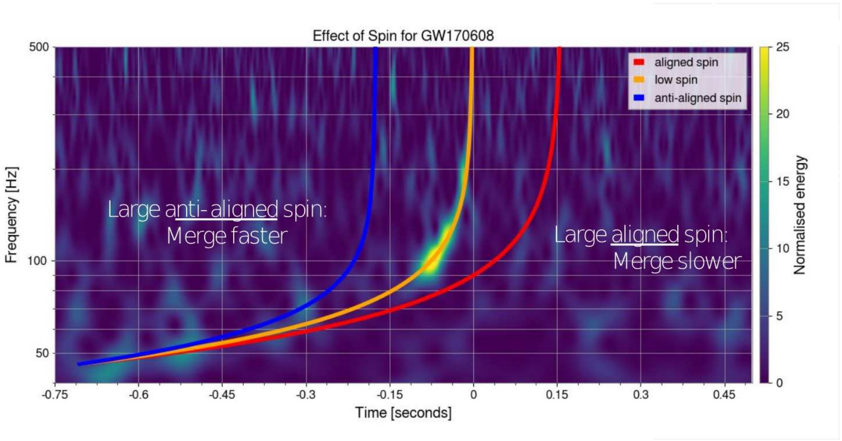
expresses the signal-to-noise (SNR) of the best-matched signal buried in the data.





Allen et al. Phys. Rev. D 85 122006 (2012)

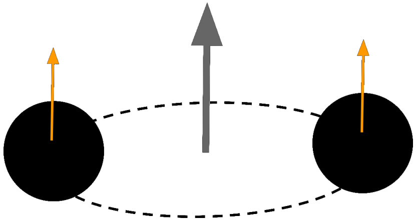
Impact of parameters on the waveform



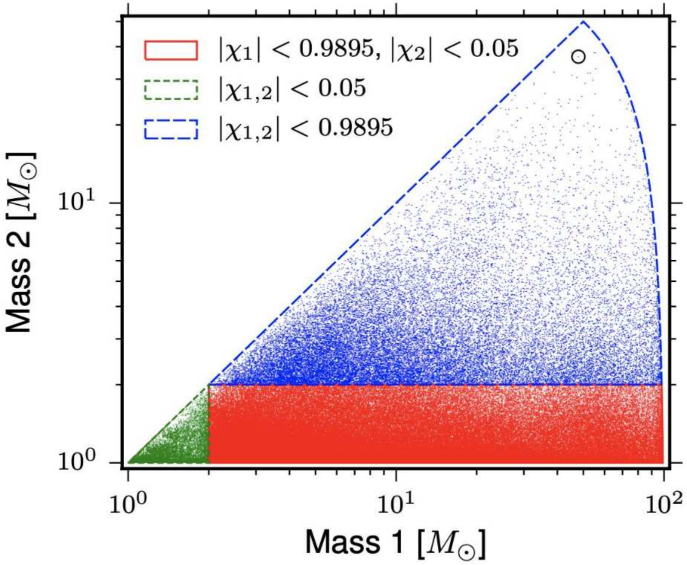
(see the `pyCBC` tutorial:
`http://pycbc.org/pycbc/latest/html/waveform.html#plotting-frequency-evolution-of-td-waveform`)

Simplistic signal model (for detection only)

- ⋆ components are black holes ($\rightarrow$ no tidal effects),
- ⋆ No in-plane spins ($\rightarrow$ no precession, spin coupling etc.),
- ⋆ Restrict to the dominant mode of the signal (quadrupole).



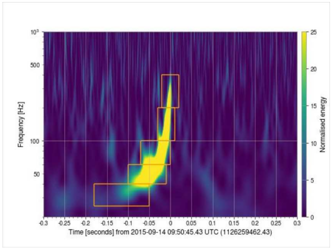
Template bank

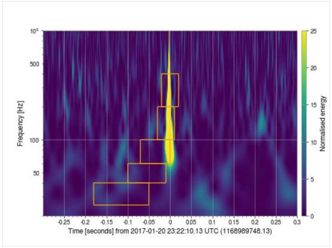


Signal consistency

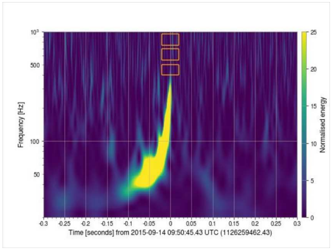
Allen, Phys. Rev. D 71, 062001 (2005): ”The χ^2 test first arose from considering a set of matched filters in different bands, and testing to see if the filter outputs all peaked at the correct time.”

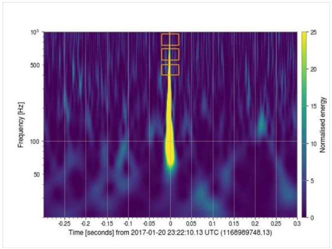
Power distribution vs time:





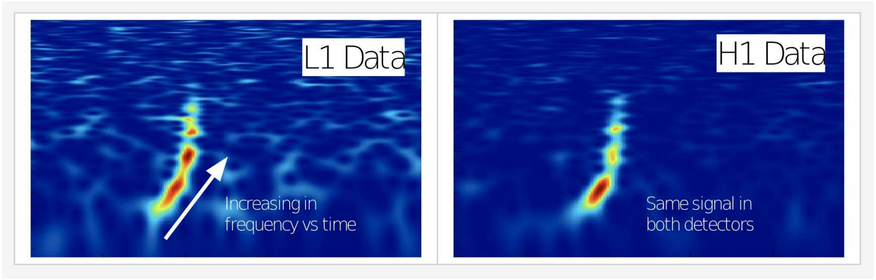
Final frequency:





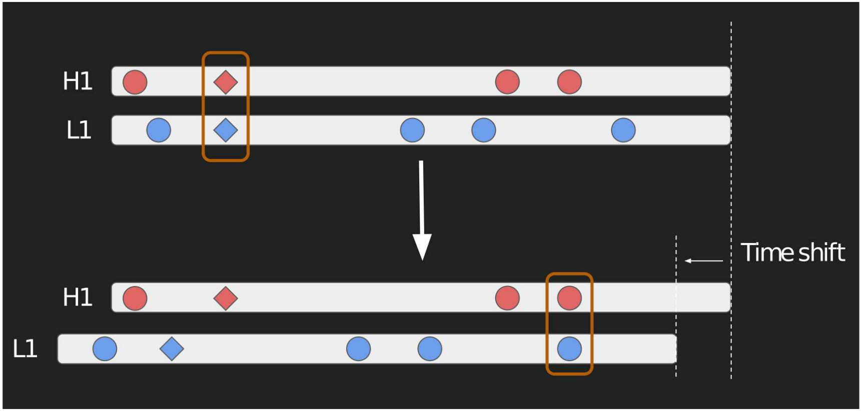
Signal consistency in unmodelled searches

With one or more detectors observing at the same time, one can use the coincidence of similar signals without knowing the precise model of the signal:

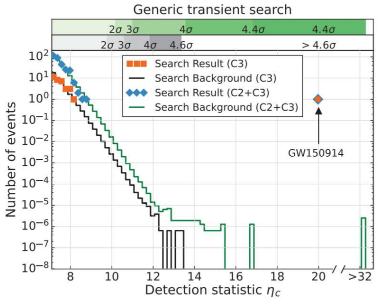


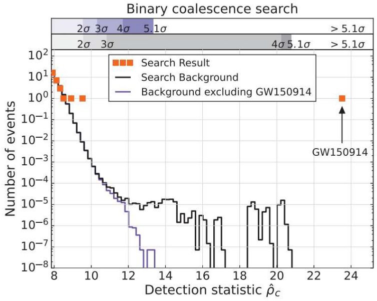
(e.g. Coherent Wave Burst: `https://gwburst.gitlab.io`, Abbott et al. Phys. Rev. D 93 122004, 2016)

Establishing significance: time shifts



Ranking and detection statistic: signal vs background





Events are ranked according to the detection statistic $\eta_c = \sqrt{2E_c/(1+E_n/E_c)}$, with

- ★ E_c: coherent signal energy obtained by cross-correlating the two reconstructed waveforms,
- ★ E_n: residual noise energy after the reconstructed signal is subtracted from the data.

Matched filter signal-to-noise ρ reweighted by taking into account the χ_r^2 statistic (consistency of the signal with the model in several frequency bands):

$$\hat{\rho}_c = \rho/((1+(\chi_r^2)^3)/2)^{1/6}.$$

(B. P. Abbott et al. Phys. Rev. Lett. 116 061102, 2016)

Literature / resources

- ⋆ GW Open Science Center `https://www.gw-openscience.org/about`
- ⋆ GW Open Science Center Tutorials
  `https://www.gw-openscience.org/tutorials`
- ⋆ LIGO - Virgo Collaboration GW Open Data Workshop #3
  `https://www.gw-openscience.org/s/workshop3`
- ⋆ `GWpy`, a python package for gravitational-wave astrophysics
  `https://gwpy.github.io`
- ⋆ `PyCBC`, free and open software to study gravitational waves
  `https://pycbc.org`