Gravitational Wave Data Analysis

2021 summer School on Numerical Relativity and Gravitational Waves

16 August ~ 20 August, 2021

Hyung Won Lee(Inje University)

with Jeongcho Kim(Inje), Chunglee Kim(Ewha)

Contents

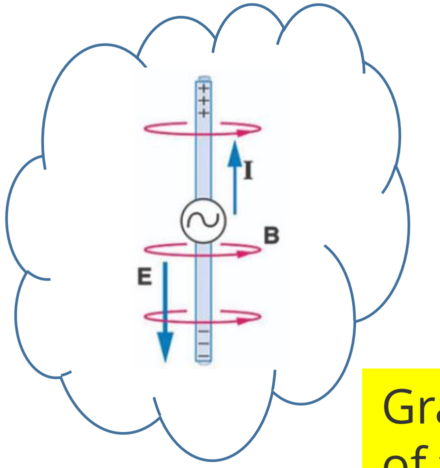
- What is gravitational wave?
- Plane wave solution
- Geometry of detector
- Gravitational wave detections
- How data obtained
- Data Analysis
- Time series data and Nyquist frequency
- Work in frequency domain

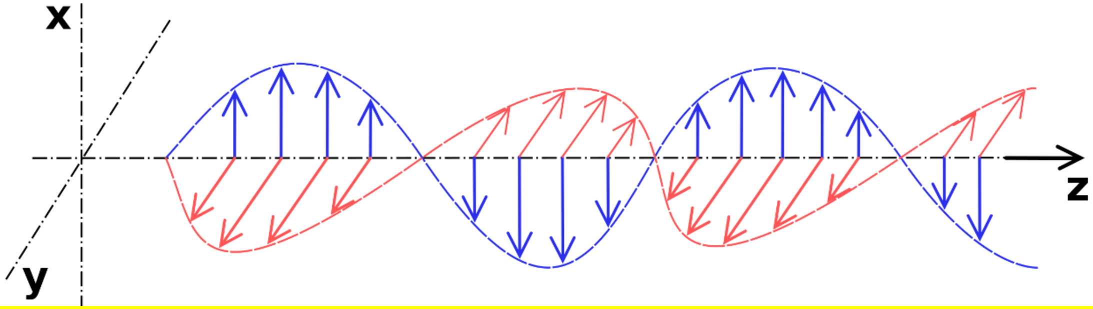
Contents

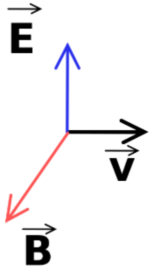
- CBC signal
- Parameter estimation
- Bayesian inference
- Various PE methods
- Questions

Gravitational Waves

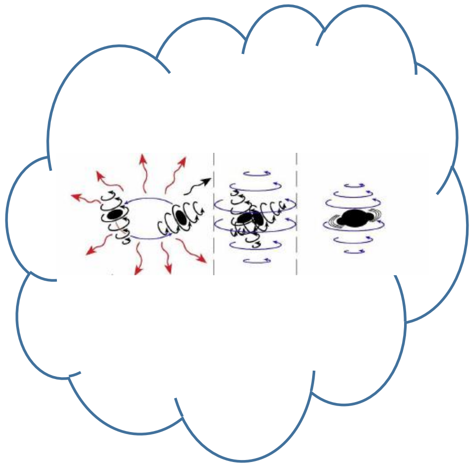
What is Gravitational Wave?

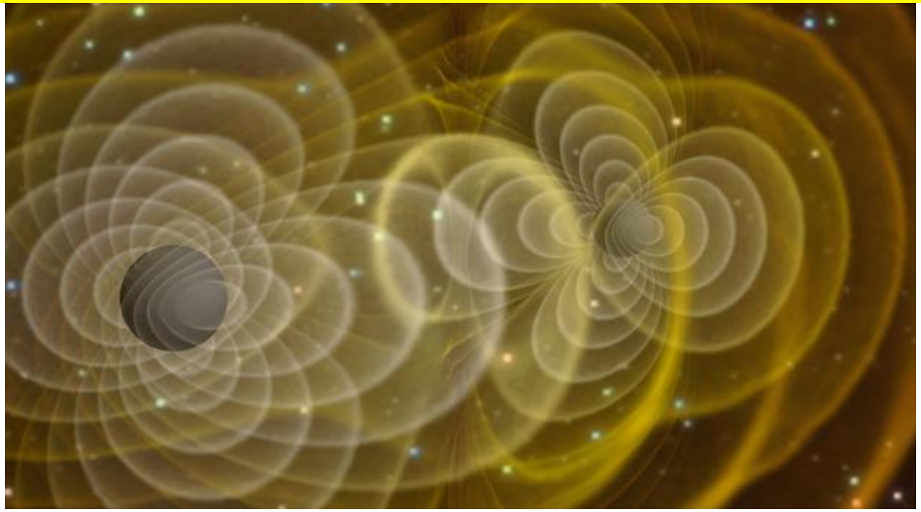


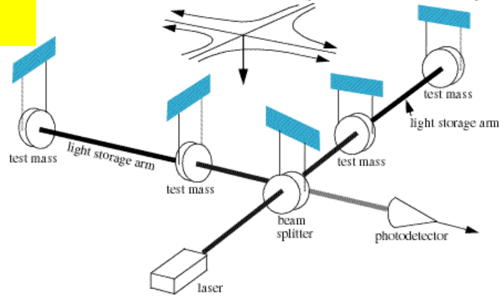




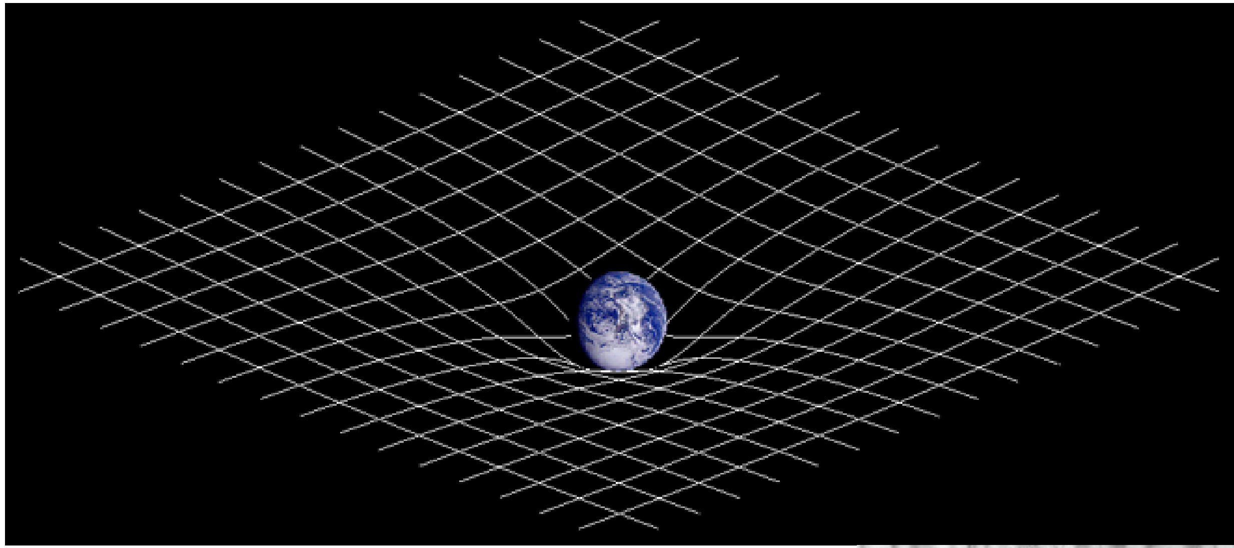
Gravitational waves are 'ripples' in space-time caused by some of the most violent and energetic processes in the Universe

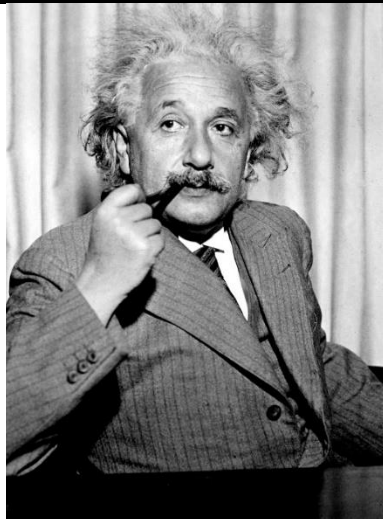






Gravitational Waves



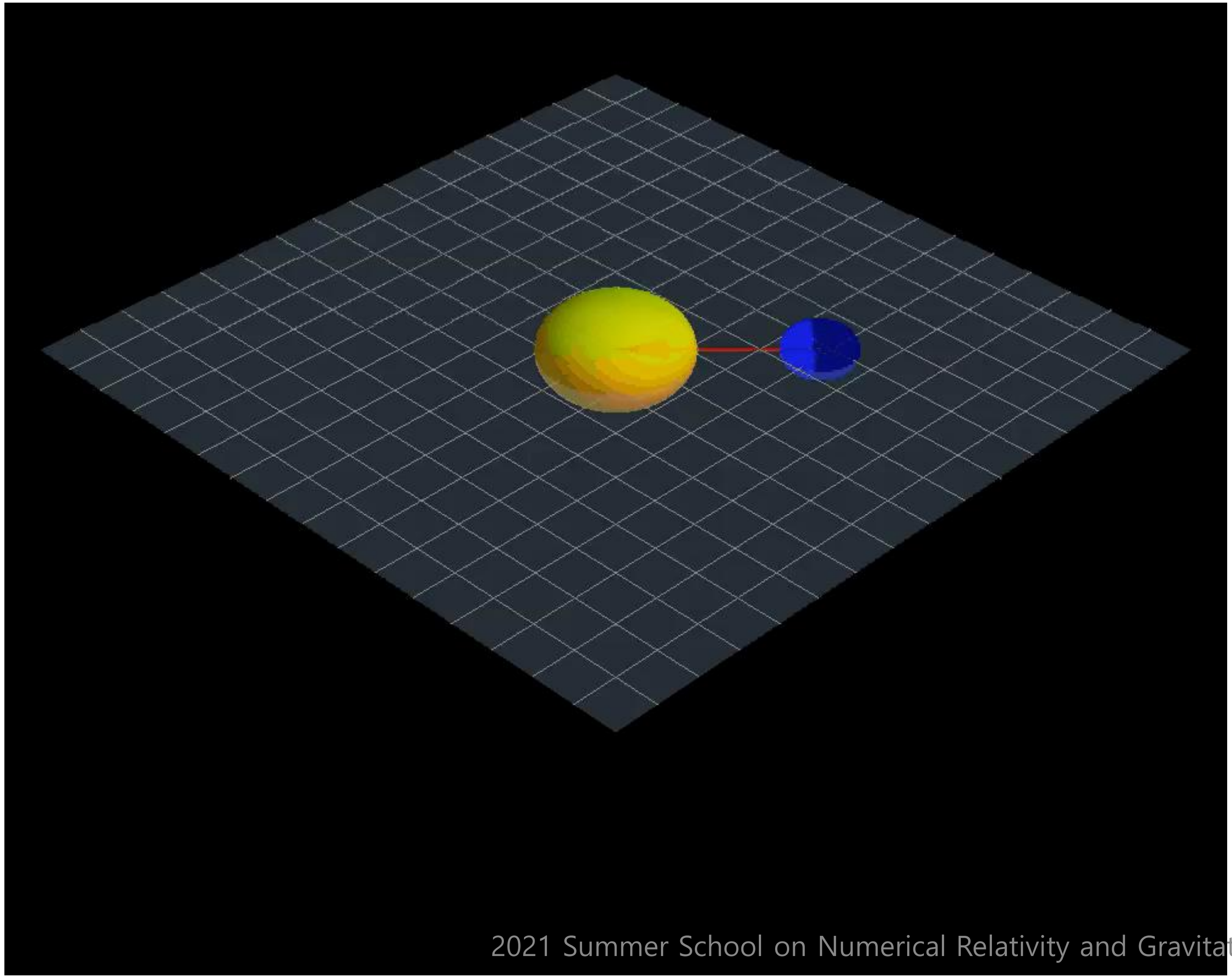


- Linearized Einstein Equation
- $g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu},\ \left|h_{\mu\nu}\right| \ll 1$
- $G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$
- $\eta^{\alpha\beta} \partial_\alpha \partial_\beta h_{\mu\nu} + O(h^2) = -\frac{16\pi G}{c^4} T_{\mu\nu},\ \partial^\mu h_{\mu\nu} = 0$
- Wave equation
- $\eta^{\alpha\beta} \partial_\alpha \partial_\beta h_{\mu\nu} = \Box h_{\mu\nu} = \left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2\right) h_{\mu\nu} = 0$

$$h_{\mu\nu} = \frac{2G}{Rc^4} \ddot{I}_{\mu\nu}$$

Binary orbit in Newtonian Theory

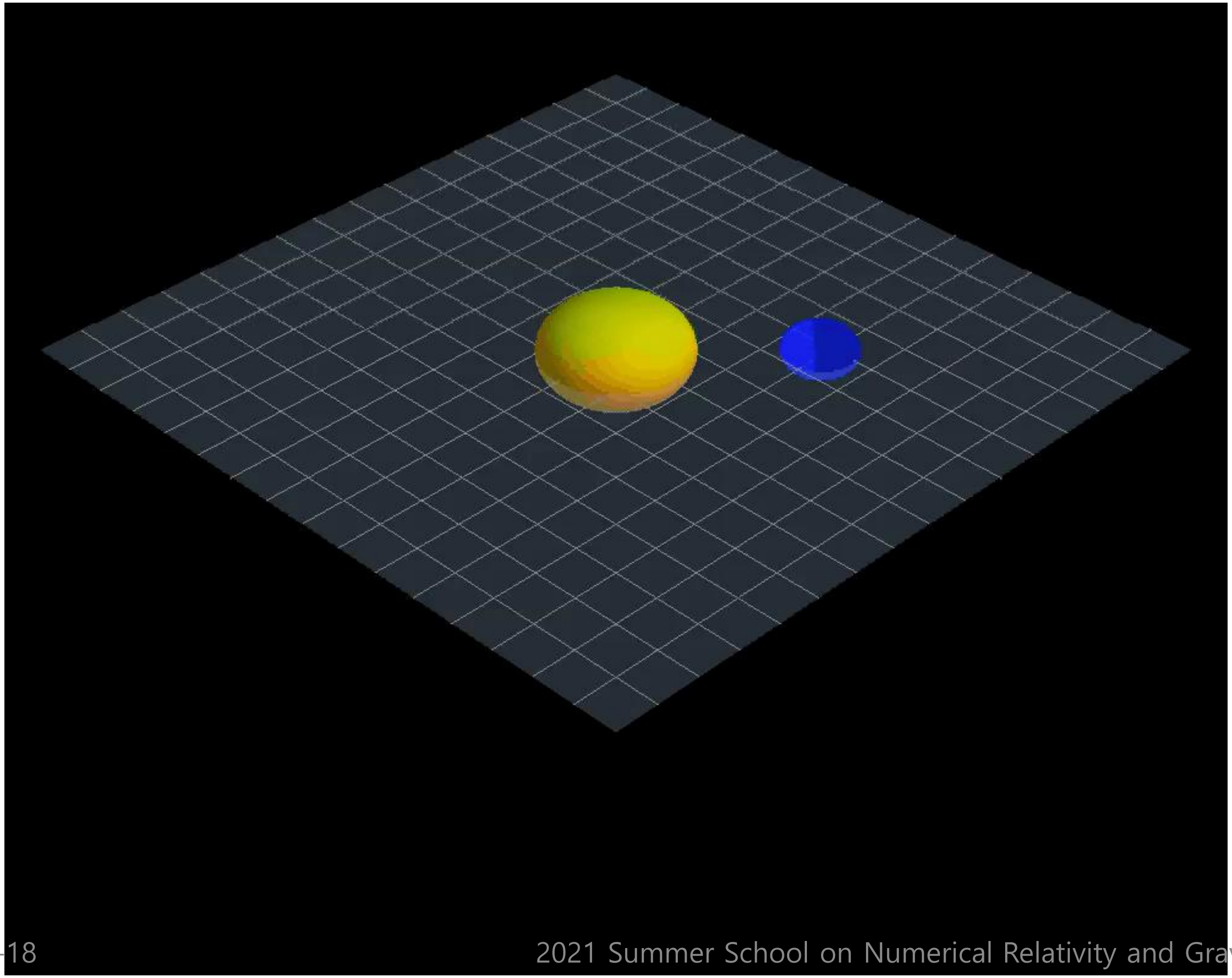
Orbital motion does not change space-time



Credit: Carl Rodriguez

General Relativity

Mass, Energy ~ curved space-time

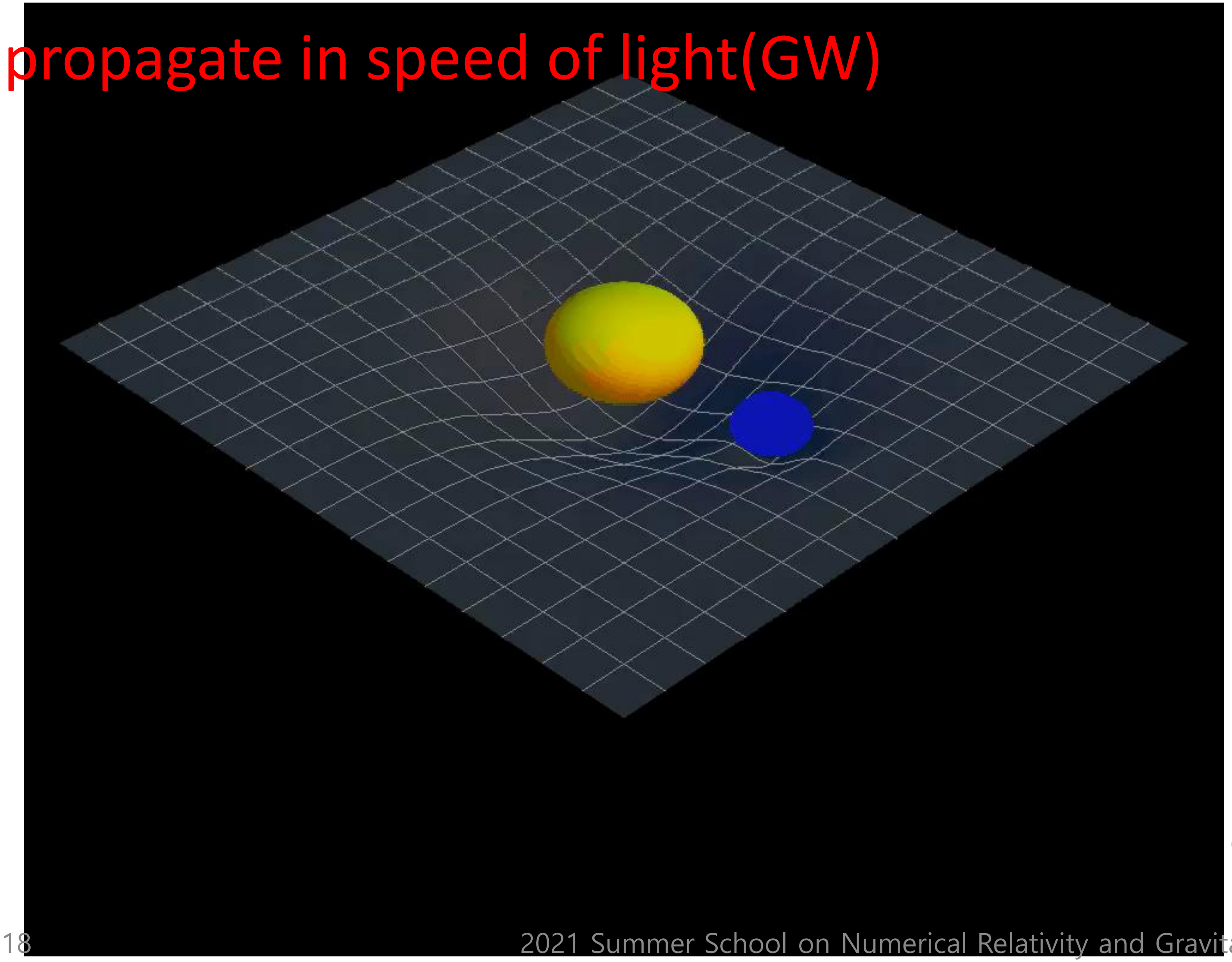


Credit: Carl Rodriguez

아인슈타인의 상대성 이론과 쌍성의 궤도 운동

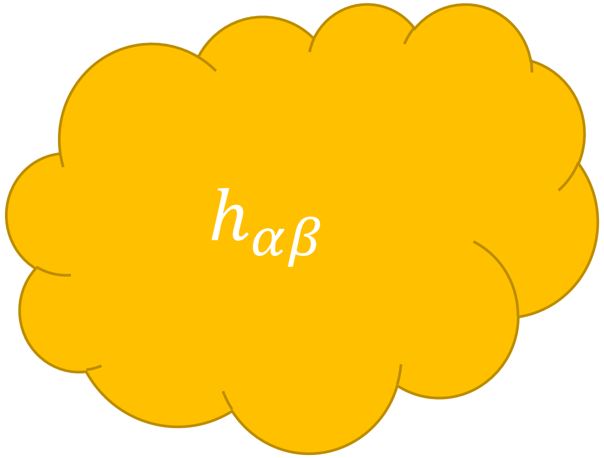
Change in space-time caused by binary motion

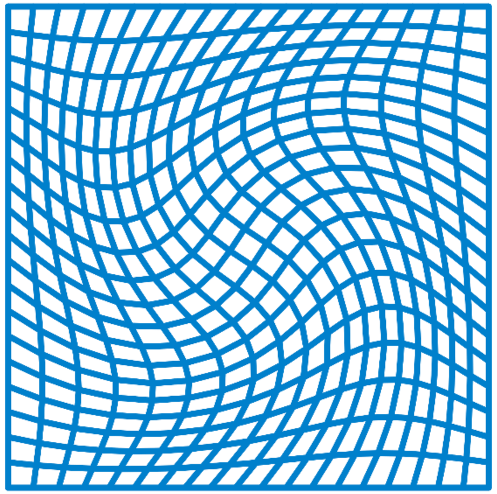
propagate in speed of light(GW)



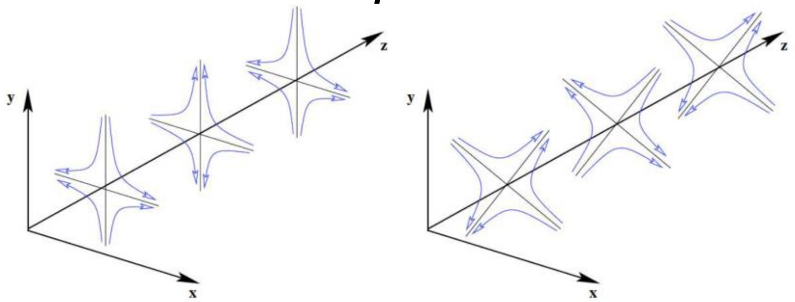
Credit: Carl Rodriguez

Degrees of Freedom





$h_{\alpha\beta}^{TT}$



Symmetric 2nd rank tensor
10 degrees of freedom

$$\Box h_{\alpha\beta} = \frac{8\pi G}{c^4}\tau_{\alpha\beta}$$

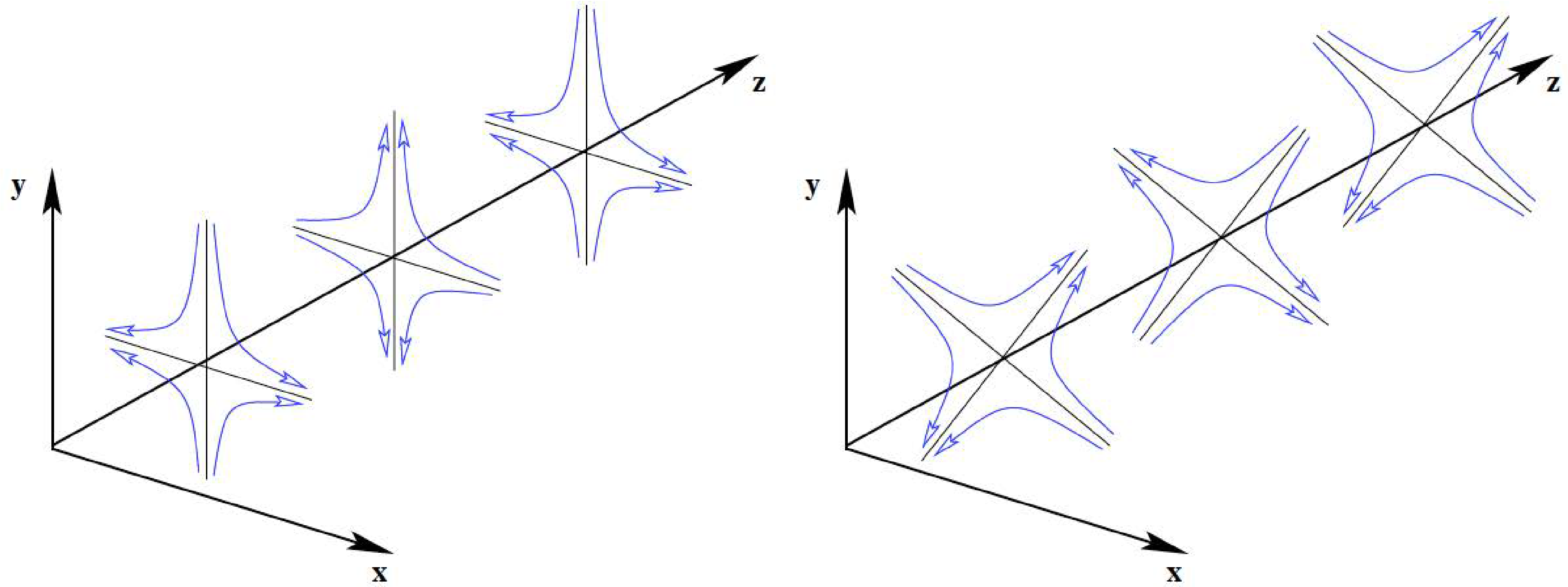
General coordinate
transformation gauge

$x^\mu \rightarrow x^\mu + \xi^\mu$
Lorentz gauge condition
$\partial_\alpha h^{\alpha\beta} = 0$ (4 conditions)
10 – 4 = 6 degrees of freedom

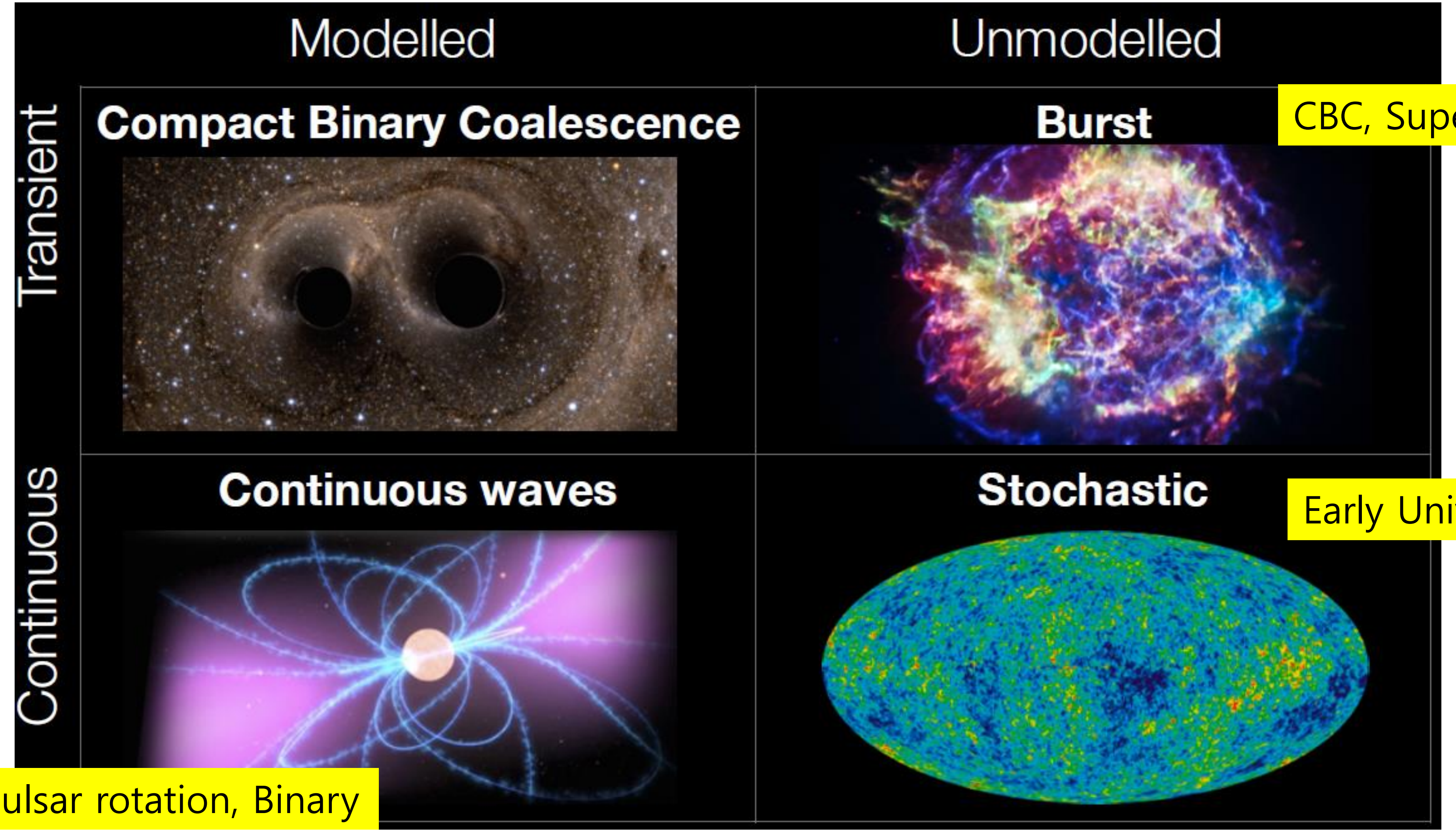
Gauge field selection freedom
Spatial and traceless
$\xi^\mu = B^\mu e^{ik_\alpha x^\alpha}, k_\alpha k^\alpha = 0$(4 condition)
6 – 4 = 2 degrees of freedom

GWs and lines of force

Lines of force for h_{+}^{TT} and $h_{\times}^{\mathrm{TT}}$



GW Sources

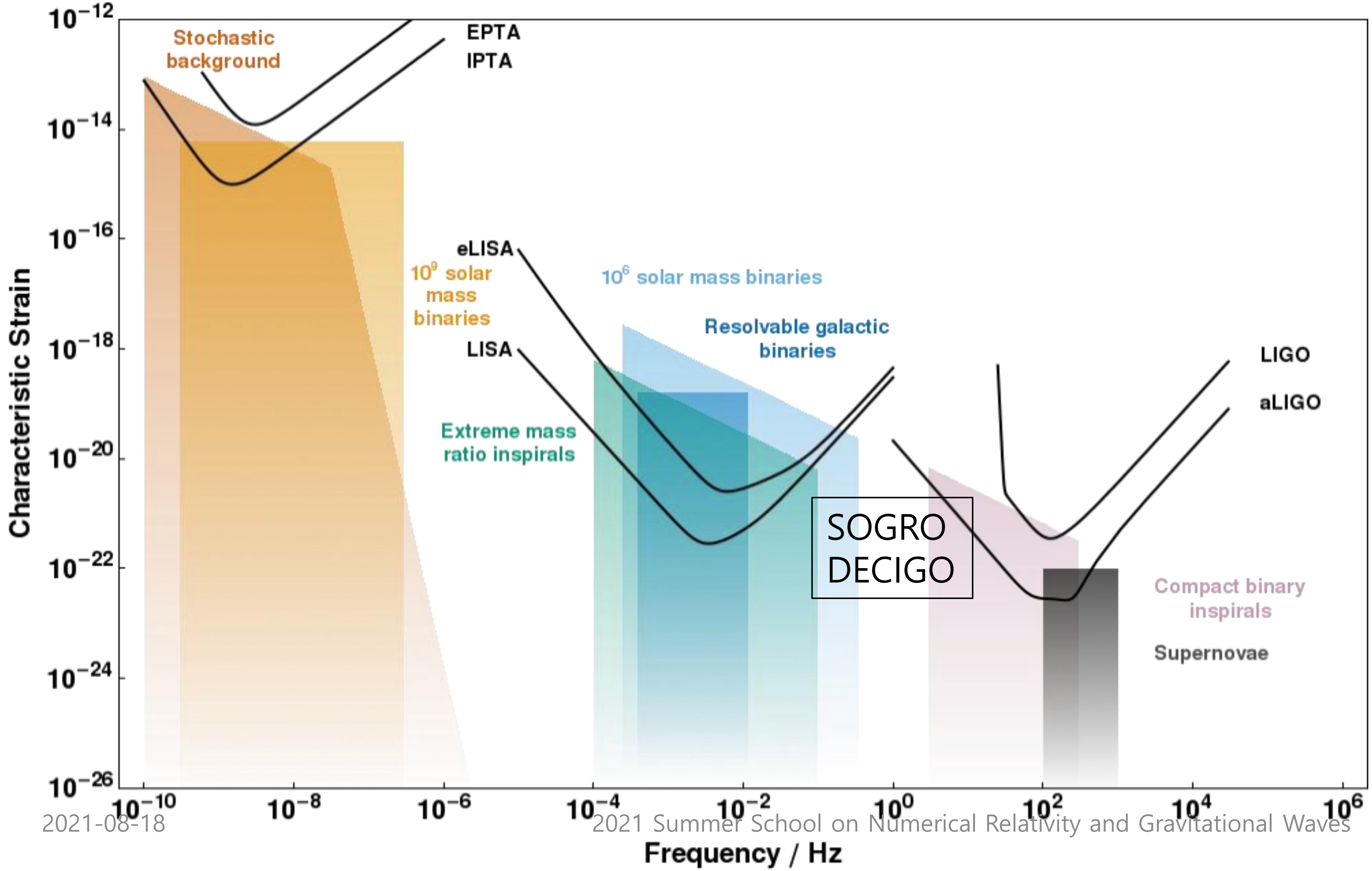


CBC, Supernovae, GRB, Pulsar glitch

Early Universe, Astronomical origin

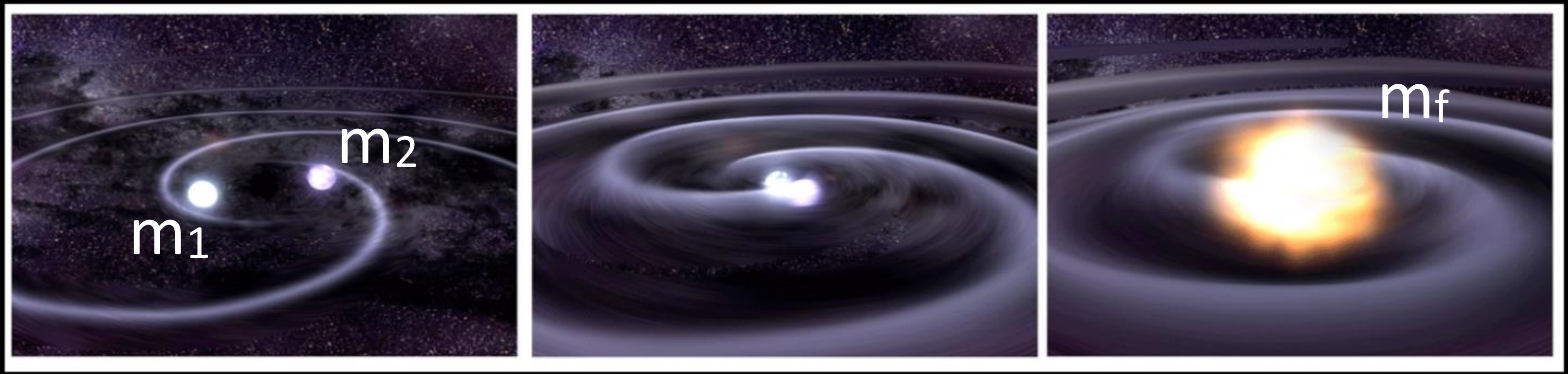
Pulsar rotation, Binary

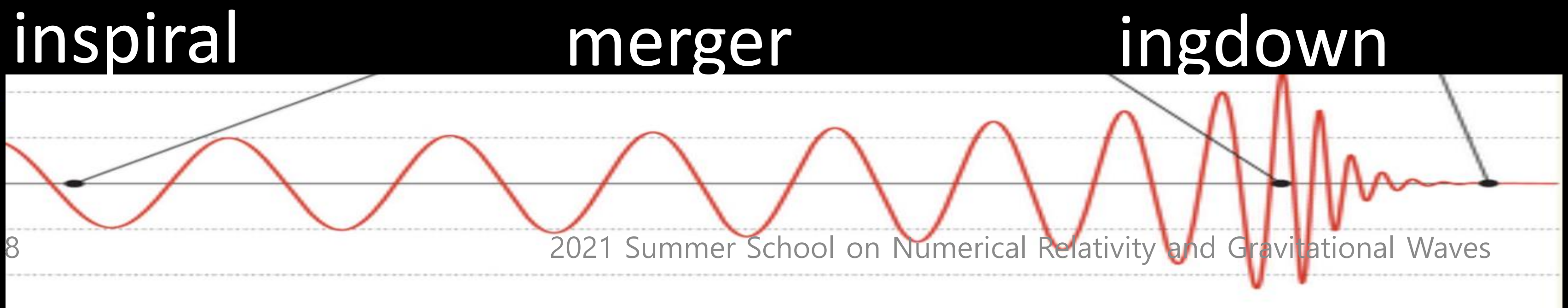
Detector Bands



Main GW sources:
Compact Binary Coalescences (CBCs)

$m_1+m_2 \sim m_f$ + GWs





Gravitational Wave

$$g_{\mu\nu} \cong \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1+h_{11} & h_{12} & 0 \\ 0 & h_{12} & 1-h_{11} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \frac{8\pi G}{c^4} T_{\mu\nu}$$

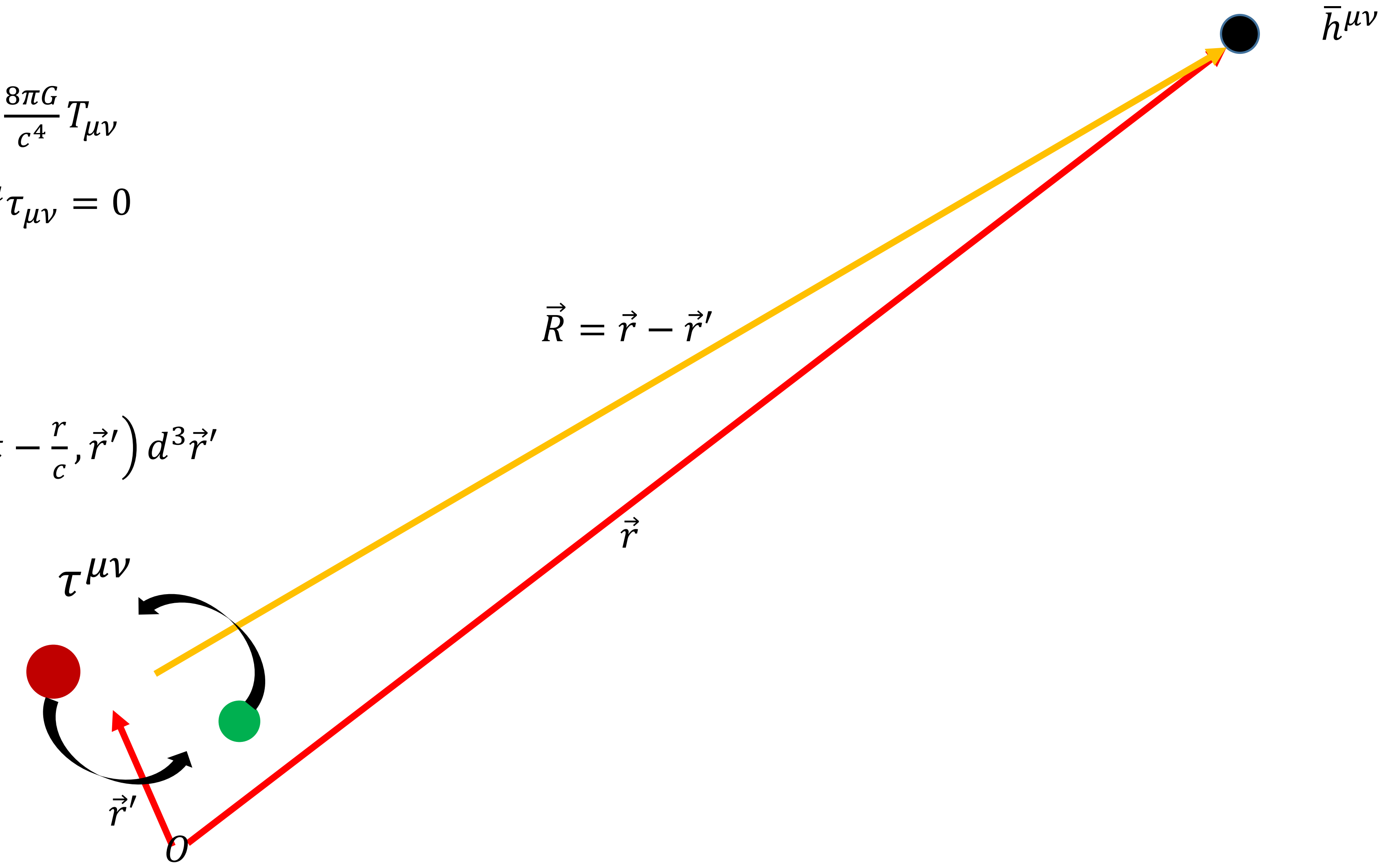
$$\Box \bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4} \tau_{\mu\nu},\ \partial^\mu \tau_{\mu\nu} = 0$$

$$\|\vec{r} - \vec{r}'\| \cong r,\ r \ll r'$$

$$t - \frac{\|\vec{r} - \vec{r}'\|}{c} \cong t - \frac{r}{c}$$

$$\bar{h}^{\mu\nu}(t, \vec{r}) \cong \frac{4G}{c^4 r} \int \tau^{\mu\nu}\left(t - \frac{r}{c}, \vec{r}'\right) d^3\vec{r}'$$

$$h_{ij}^{TT} \cong \frac{2G}{c^4 r} \ddot{I}_{ij}^{TT}\left(t - \frac{r}{c}\right)$$



Plane Wave Solution

Solution in vacuum

- $\Box \bar{h}_{\mu\nu} = 0,\ \partial^{\mu}\bar{h}_{\mu\nu} = 0$
- $\bar{h}_{\mu\nu} = A_{\mu\nu}\cos\left(k_{\alpha}x^{\alpha} - \phi_{(\mu)(\nu)}\right), \eta^{\alpha\beta}k_{\alpha}k_{\beta} = 0$
- $x^{\alpha} = \{ct, x, y, z\},\ k^{\alpha} = \left\{\frac{\omega}{c}, k^{x}, k^{y}, k^{z}\right\}$
- $k_{\alpha}x^{\alpha} = -\omega t + \vec{k}\cdot\vec{x}, \omega = 2\pi f = c\left|\vec{k}\right|$
- $\bar{h}_{\mu\nu} = A_{\mu\nu}\cos\left(\omega t - \vec{k}\cdot\vec{x} + \phi_{(\mu)(\nu)}\right), \bar{h}_{\mu\nu}k^{\nu} = 0$ This equations reduce 10 to 6
- $\xi^{\alpha}(t, \vec{x}) = B^{\alpha}\cos\left(\omega t - \vec{k}\cdot\vec{x} + \phi_{(\alpha)}\right)$
- $x'^{\mu} = x^{\mu} + \xi^{\mu}$

Transverse Traceless coordinate

- $\bar{h}'_{\mu\nu}U'^{\nu} = 0, \eta^{\mu\nu}\bar{h}'_{\mu\nu} = 0, \bar{h}'_{\mu\nu}k'^{\nu} = 0, \bar{h}'_{\mu\nu} = h'_{\mu\nu}$

TT gauge

Trace reversed and original is same in TT gauge

Global Lorentz transformation $x'^{\alpha} = \Lambda^{\alpha}_{\beta}x^{\beta}$ preserve all these properties.

Choose coordinate to be $U^{\mu} = \frac{dx^{\mu}}{dt} = (1,0,0,0)$ $\Rightarrow$ $h_{\mu 0} = 0$

Rotate coordinate to be $k^{\mu} = \left(\frac{\omega}{c}, 0, 0, \frac{\omega}{c}\right)$ $\Rightarrow$ $h_{\mu 3} = 0$

$h_{11} + h_{22} = 0$

$$h_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_{11} & h_{12} & 0 \\ 0 & h_{12} & -h_{11} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Transverse Traceless coordinate

- $h_+ \equiv h_{11} = -h_{22}, h_\times \equiv h_{12} = h_{21}$
- $h_{\mu\nu}^{TT}(t,\vec{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+(t,\vec{x}) & h_\times(t,\vec{x}) & 0 \\ 0 & h_\times(t,\vec{x}) & -h_+(t,\vec{x}) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+\left(t-\frac{z}{c}\right) & h_\times\left(t-\frac{z}{c}\right) & 0 \\ 0 & h_\times\left(t-\frac{z}{c}\right) & -h_+\left(t-\frac{z}{c}\right) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
- $h_+(t,\vec{x}) = A_+ \cos\left[\omega\left(t-\frac{z}{c}\right) + \phi_+\right]$
- $h_\times(t,\vec{x}) = A_\times \cos\left[\omega\left(t-\frac{z}{c}\right) + \phi_\times\right]$
- $h_{\mu\nu}^{TT}(t,\vec{x}) = h_+(t,\vec{x})e_{\mu\nu}^+ + h_\times(t,\vec{x})e_{\mu\nu}^\times$
- $R_{x0x0}^{TT} = -R_{y0y0}^{TT} = -R_{x0xz}^{TT} = R_{y0yz}^{TT} = R_{xzxz}^{TT} = -R_{yzyz}^{TT} = -\frac{1}{2}\partial_0^2 h_+$
- $R_{x0y0}^{TT} = -R_{x0yz}^{TT} = -R_{y0xz}^{TT} = R_{xzyz}^{TT} = -\frac{1}{2}\partial_0^2 h_\times$

$$e_{\mu\nu}^+ = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ e_{\mu\nu}^\times = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Observer's proper frame

- $ds^2 = -c^2 d\hat{t}^2 + \delta_{ij} d\hat{x}^i d\hat{x}^j + \mathcal{O}(\hat{x}^2)$
- $\hat{x}^\alpha_{\mathcal{A}}(\hat{t}) = (c\hat{t}, 0,0,0)$
- $\hat{x}^\alpha_{\mathcal{B}}(\hat{t}) = \left(c\hat{t}, \hat{x}^1(\hat{t}), \hat{x}^2(\hat{t}), \hat{x}^3(\hat{t})\right)$
- $\hat{\xi}^\alpha(\hat{t}) \equiv \hat{x}^\alpha_{\mathcal{B}}(\hat{t}) - \hat{x}^\alpha_{\mathcal{A}}(\hat{t}) = \left(0, \hat{x}^1(\hat{t}), \hat{x}^2(\hat{t}), \hat{x}^3(\hat{t})\right)$
- $\frac{D^2 \hat{\xi}^\alpha}{d\hat{t}^2} = -c^2 \hat{R}^\alpha{}_{\beta\gamma\delta} \hat{u}^\beta \hat{\xi}^\gamma \hat{u}^\delta, \hat{u}^\alpha = \frac{d\hat{x}^\alpha}{c d\hat{t}} = (1,0,0,0)$
- $\frac{d^2 \hat{x}^i}{d\hat{t}^2} = -c^2 \hat{R}^i{}_{0j\delta} \hat{x}^j = -c^2 \hat{R}_{i0j0} \hat{x}^j + \mathcal{O}(\hat{x}^3)$
- $\hat{R}_{i0j0} = \hat{R}^{TT}_{i0j0} + \mathcal{O}(h^2)$

Observer's proper frame

- $\frac{d^2\hat{x}^i}{d\hat{t}^2} = \frac{1}{2}\frac{\partial^2 h_{ij}^{TT}}{\partial\hat{t}^2}\hat{x}^j, \hat{x}^i(\hat{t}) = \hat{x}_0^i, \frac{d\hat{x}^i}{d\hat{t}} = 0$ Initial condition
- $\hat{x}^i(\hat{t}) = \hat{x}_0^i + \mathcal{O}(h)$
- $\frac{d^2\hat{x}^i}{d\hat{t}^2} = \frac{1}{2}\frac{\partial^2 h_{ij}^{TT}}{\partial\hat{t}^2}\hat{x}_0^i$
- $\hat{x}^i(\hat{t}) = \left(\delta_{ij} + \frac{1}{2}h_{ij}^{TT}(\hat{t})\right)\hat{x}_0^i$ Solution in linear order

$$\hat{x}(\hat{t}) = \hat{x}_0 + \frac{1}{2}\Big(h_+(\hat{t})\hat{x}_0 + h_\times(\hat{t})\hat{y}_0\Big), \tag{1.83a}$$

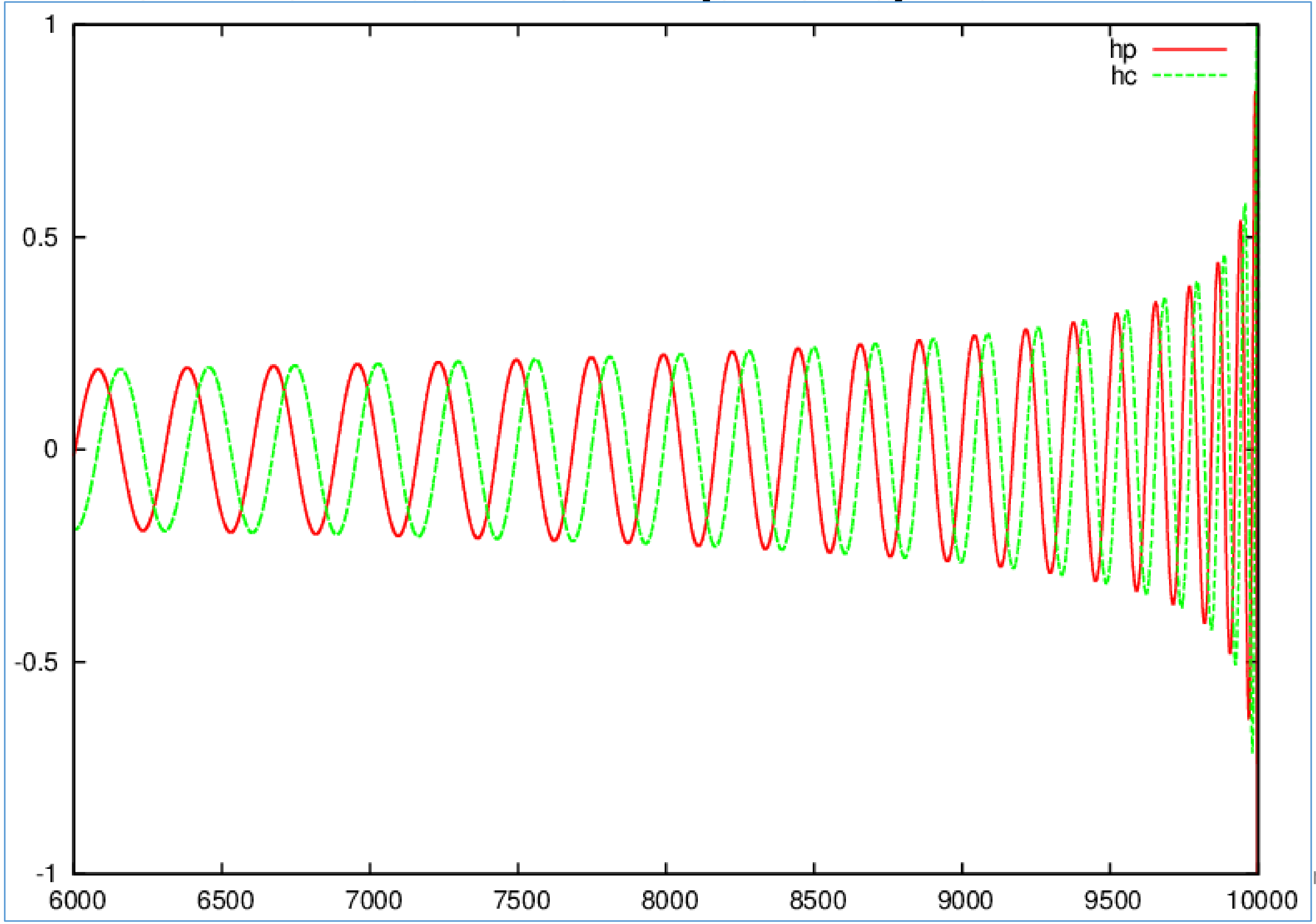
$$\hat{y}(\hat{t}) = \hat{y}_0 + \frac{1}{2}\Big(h_\times(\hat{t})\hat{x}_0 - h_+(\hat{t})\hat{y}_0\Big), \tag{1.83b}$$

$$\hat{z}(\hat{t}) = \hat{z}_0. \tag{1.83c}$$

Theoretical Waveforms

- Known only for the very restricted cases, CBC inspiral
- Newtonian Chirp signal from binary system
  - $h_+(t) = -\frac{GM_c}{c^2 r}\frac{1+\cos^2\iota}{2}\left(\frac{c^3(t_c-t)}{5GM_c}\right)^{-\frac{1}{4}}\cos\left[2\varphi_c - 2\left(\frac{c^3(t_c-t)}{5GM_c}\right)^{\frac{5}{8}}\right]$
  - $h_\times(t) = -\frac{GM_c}{c^2 r}\cos\iota\left(\frac{c^3(t_c-t)}{5GM_c}\right)^{-\frac{1}{4}}\sin\left[2\varphi_c - 2\left(\frac{c^3(t_c-t)}{5GM_c}\right)^{5/8}\right]$

Newtonian Chirp Signal



Post-Newtonian waveform

$$
\begin{aligned}
\tilde{h}(f) &= \frac{M\,\nu}{D_L}\sum_{n=0}^{5}\sum_{k=1}^{7} V_k^{2+n}\left(k\,\frac{dF}{dt}\right)^{-1/2} \\
&\times\left(\alpha_k^{(n)}\,e^{i\,(2\,\pi\,f\,t(F)-k\,\Psi(F)\,-\,\pi/4)}+\beta_k^{(n)}\,e^{i(2\,\pi\,f\,t(F)\,-(k\,\Psi(F)-\pi/2)-\pi/4)}\right)\,, \\
&= \frac{M\,\nu}{D_L}\sum_{n=0}^{5}\sum_{k=1}^{7} V_k^{n-\frac{7}{2}}\sqrt{\frac{5\,\pi}{k\,48\,\nu}}\,M\,\left(1+\mathcal{S}_2\,V_k^2+\mathcal{S}_3\,V_k^3+\mathcal{S}_4\,V_k^4+\mathcal{S}_5\,V_k^5\right) \\
&\times(\alpha_k^{(n)}+e^{i\,\pi/2}\,\beta_k^{(n)})e^{i\,(k\,\Psi_{\mathrm{SPA}}(f/k)-\pi/4)}\,, \\
&= \frac{M^2}{D_L}\sqrt{\frac{5\,\pi\,\nu}{48}}\sum_{n=0}^{5}\sum_{k=1}^{7}\,V_k^{n-\frac{7}{2}}\,\mathcal{C}_k^{(n)}e^{i\,(k\,\Psi_{\mathrm{SPA}}(f/k)-\pi/4)}\,. \qquad (4.72)
\end{aligned}
$$

$$\tilde{h}(f) = \mathcal{A} f^{-7/6} e^{i\psi(f)},$$

Geometry of Detector

Wave propagation coordinate

$$ds^2 = -c^2dt^2 + (\delta_{ij} + h_{ij})dx^i dx^j$$

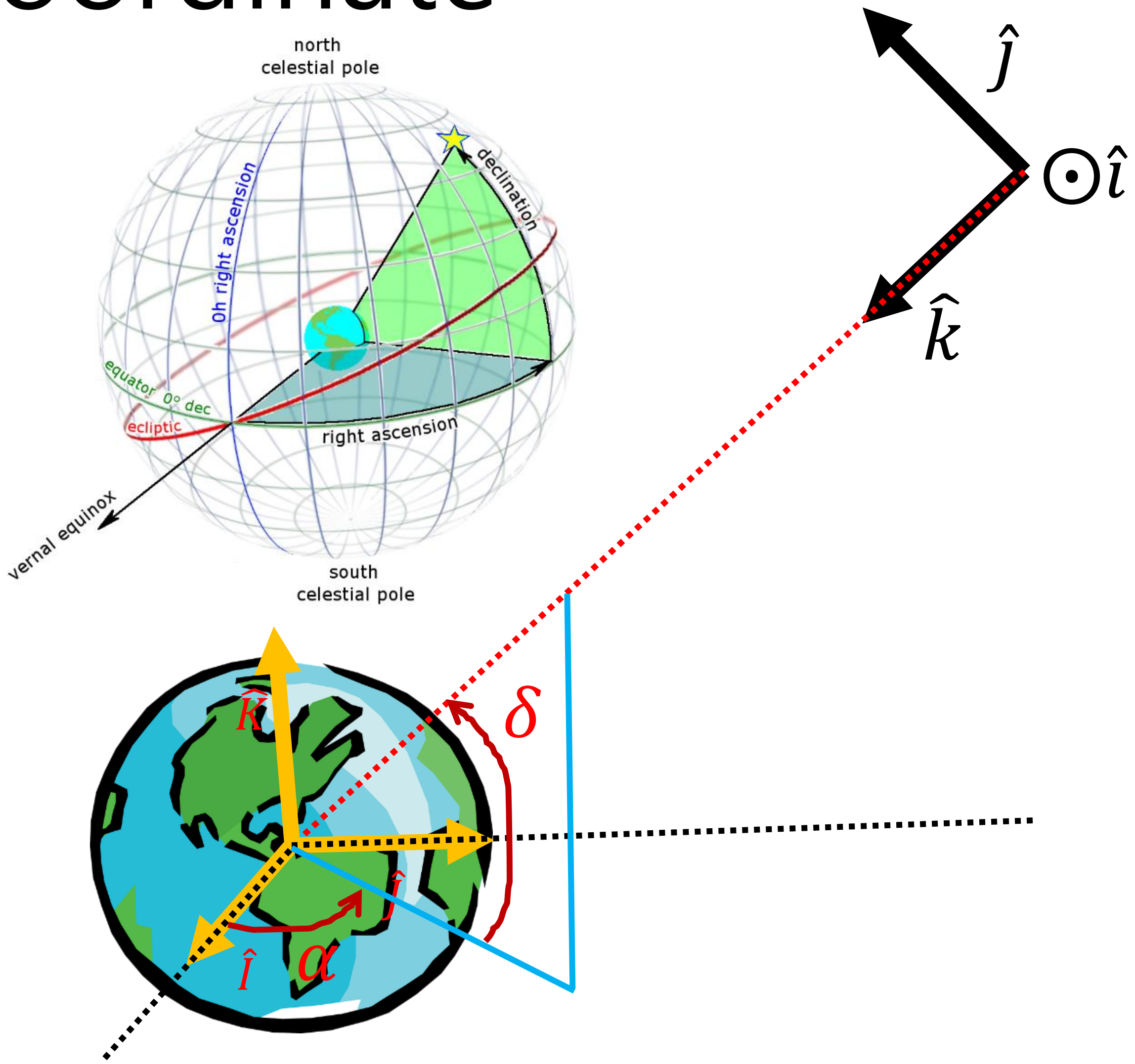
$$\left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2\right)h_{ij} = 0,\ \ h_{ij}\left(t - \frac{\vec{k}\cdot\vec{r}}{c}\right)$$

$$\vec{k} = (0,0,1), h_{ij} = h_+ e_{+ij} + h_\times e_{\times ij}$$

$$e_{+ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, e_{\times ij} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

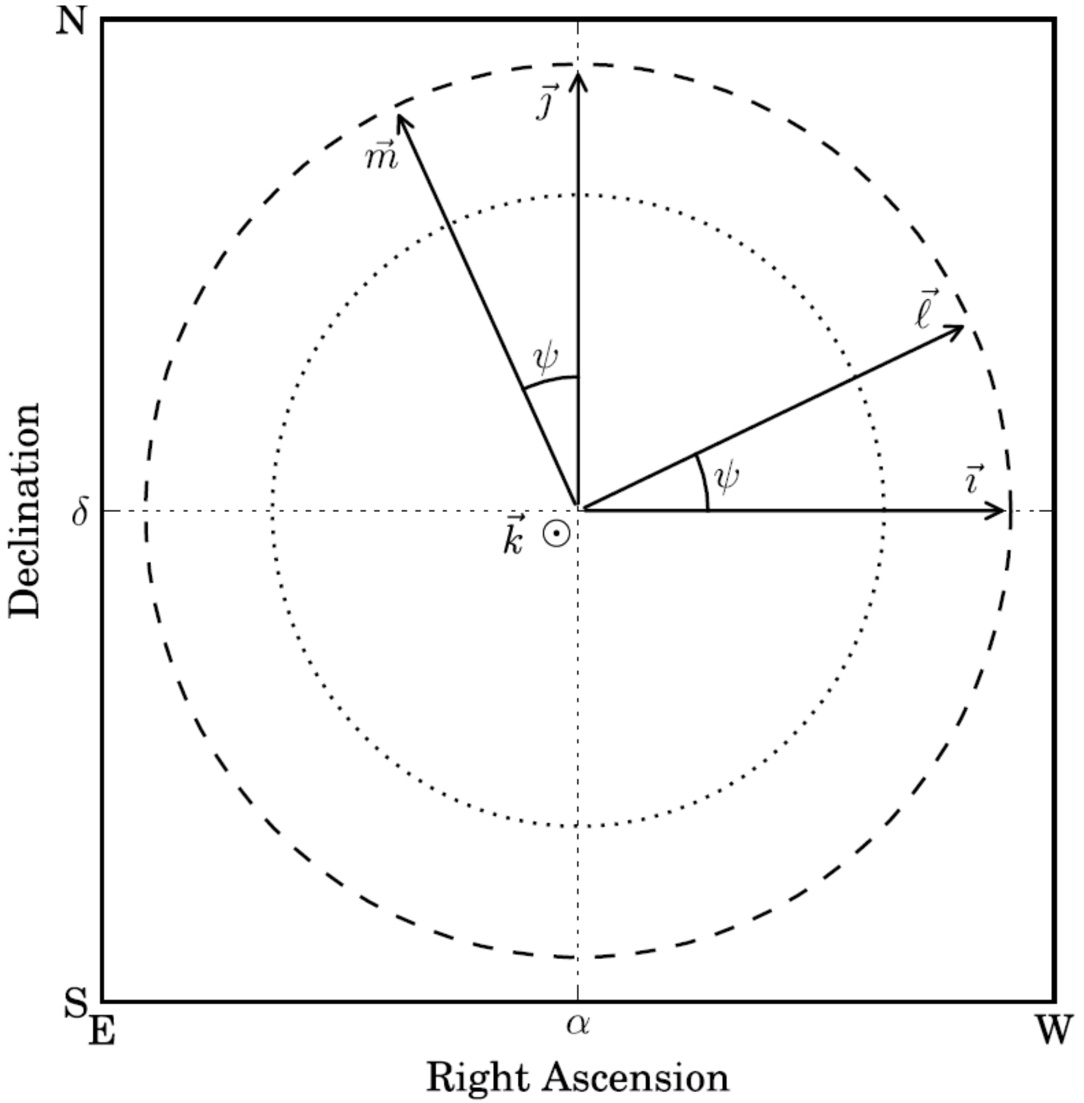
$$e_{+ij} = \ell_i \ell_j - m_i m_j, e_{\times ij} = \ell_i \ell_j + m_i m_j$$

$\vec{\ell}, \vec{m}, \vec{k}$: source's natural polarization



Direction of source

- $\vec{\ell} = \hat{\imath}\cos\psi + \hat{\jmath}\sin\psi$
- $\vec{m} = -\hat{\imath}\sin\psi + \hat{\jmath}\cos\psi$
- $\varepsilon_{+ij} = \hat{\imath}_i\hat{\imath}_j - \hat{\jmath}_i\hat{\jmath}_j, \varepsilon_{\times ij} = \hat{\imath}_i\hat{\imath}_j + \hat{\jmath}_i\hat{\jmath}_j$
- $e_{+ij} = \varepsilon_{+ij}\cos 2\psi + \varepsilon_{\times ij}\sin 2\psi$
- $e_{\times ij} = -\varepsilon_{+ij}\sin 2\psi + \varepsilon_{\times ij}\cos 2\psi$

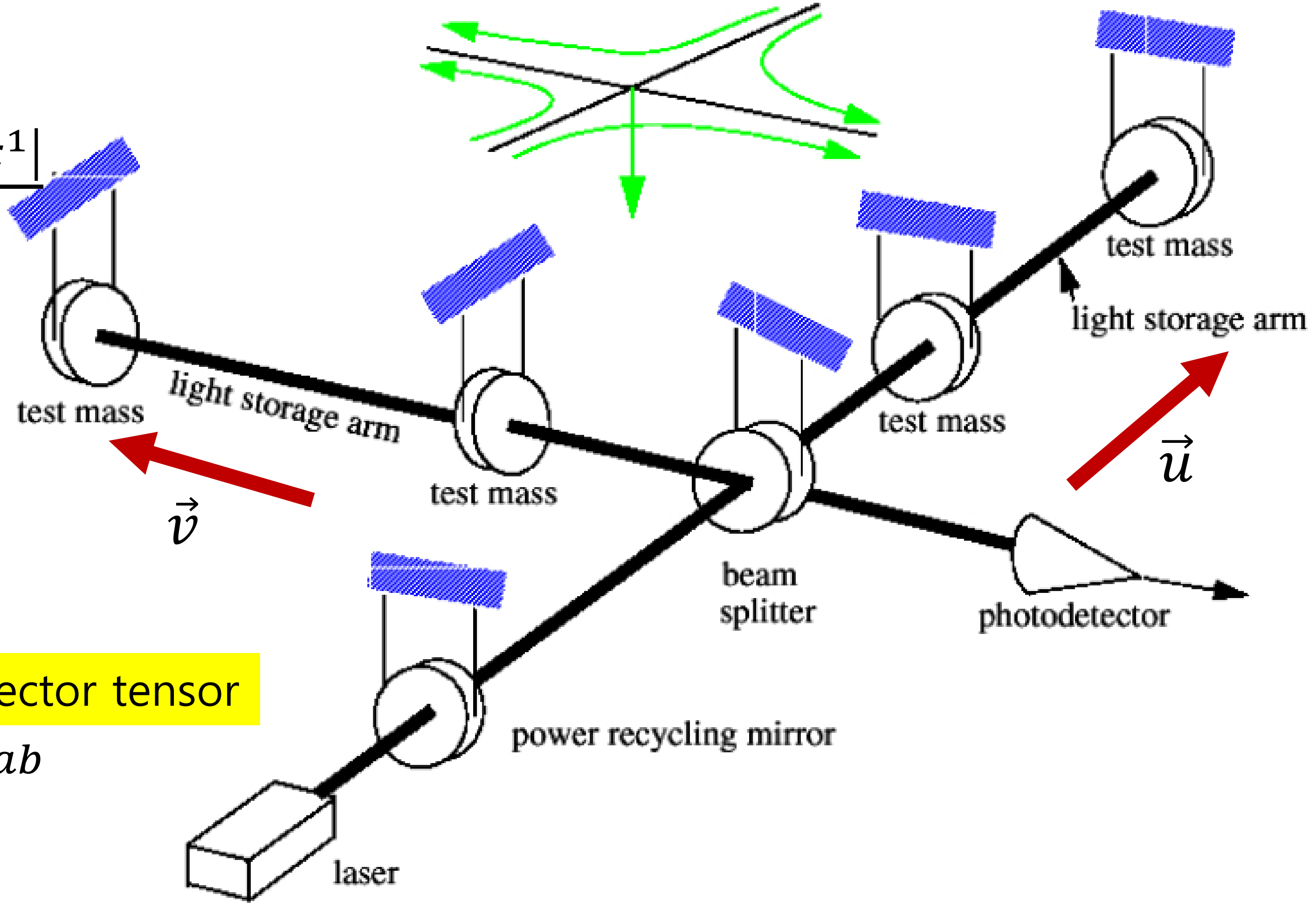


Interaction with a Detector

- $ds^2 = -c^2dt^2 + (1 + h_{11})(dx^1)^2$
- $dt = \frac{\sqrt{1+h_{11}}}{c}|dx^1| \approx \left(1 + \frac{1}{2}h_{11}\right)\frac{|dx^1|}{c}$
- $T_1 = \left(1 + \frac{1}{2}h_{11}\right)\frac{2L_0}{c} \equiv \frac{2L_1}{c}$
- $h_{11} = u^i h_{ij} u^j$
- $L_u = L_0\left(1 + \frac{1}{2}h_{ab}u^a u^b\right)$
- $L_v = L_0\left(1 + \frac{1}{2}h_{ab}v^a v^b\right)$

Detector tensor

- $h = \frac{L_u - L_v}{L_0} = h_{ab}\frac{u^a u^b - v^a v^b}{2} = h_{ab}d^{ab}$



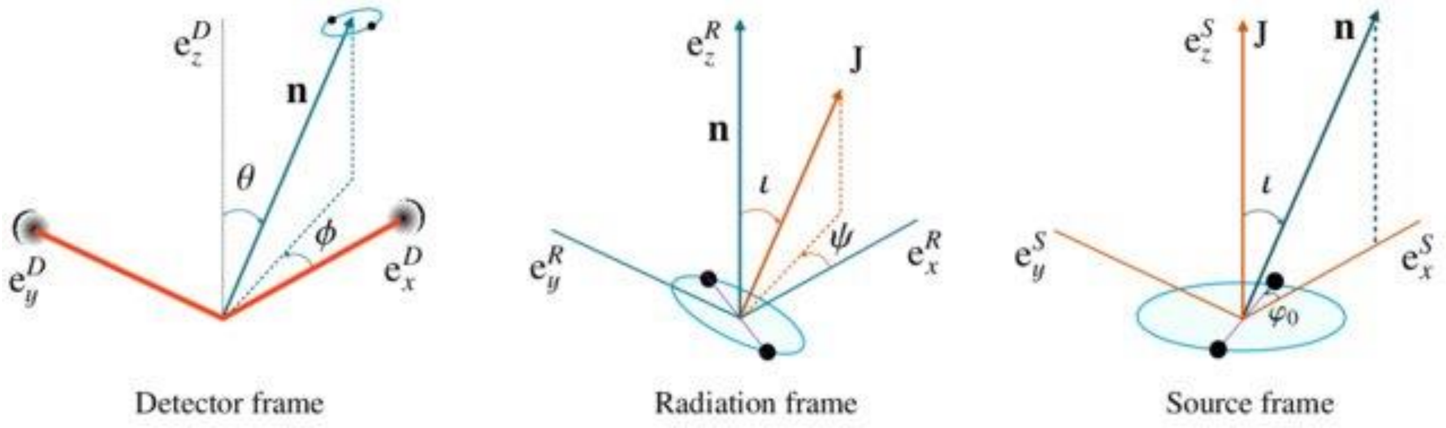
Antenna response function

- $h = h_{ab}d^{ab} = (h_+ e_{+ab} + h_\times e_{\times ab})d^{ab} = h_+ F_+ + h_\times F_\times$
- $F_+ = d^{ab} e_{+ab} = d^{ab} \varepsilon_{+ab} \cos 2\psi + d^{ab} \varepsilon_{\times ab} \sin 2\psi$
- $F_\times = d^{ab} e_{+ab} = -d^{ab} \varepsilon_{+ab} \sin 2\psi + d^{ab} \varepsilon_{\times ab} \cos 2\psi$

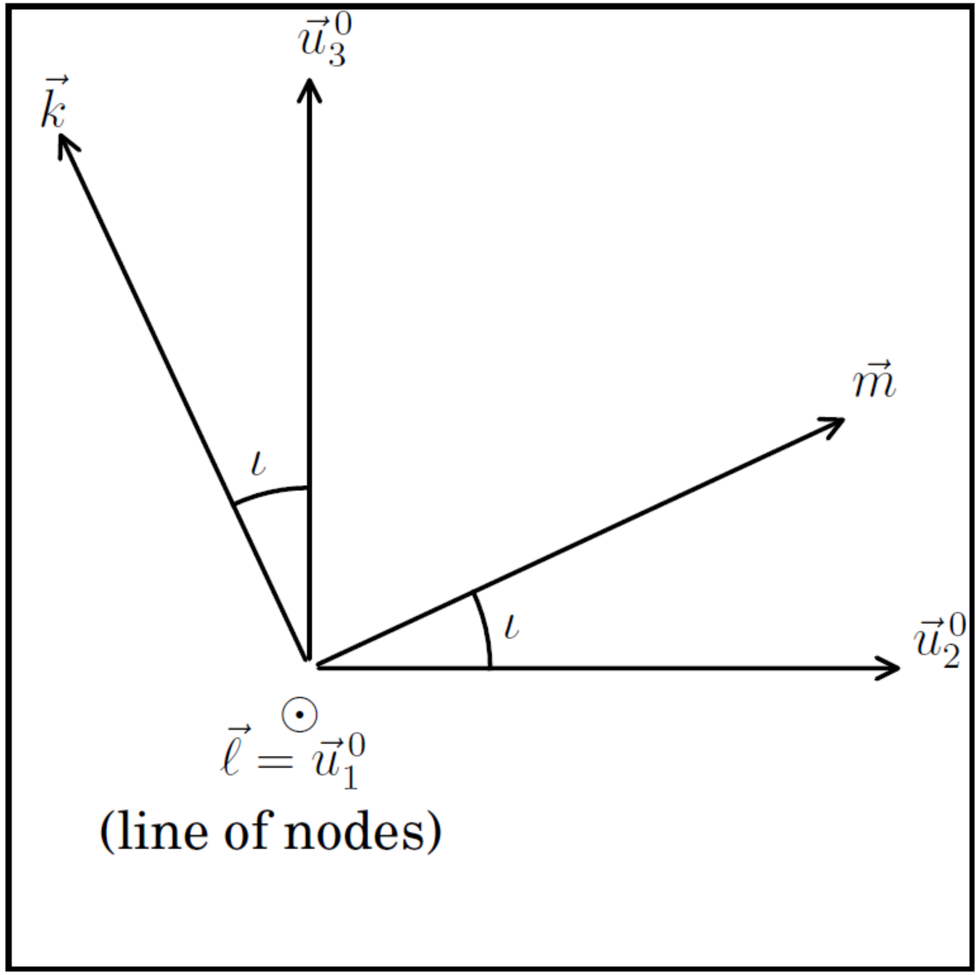
Inclination angle

$$h_+ = h_0 \frac{1+\cos^2\iota}{2} \cos[\Phi(t)+\phi_0] \qquad (3.22a)$$

$$h_\times = h_0 \cos\iota \, \sin[\Phi(t)+\phi_0] \qquad (3.22b)$$



M. Purrer, etc., https://arxiv.org/abs/1409.2349



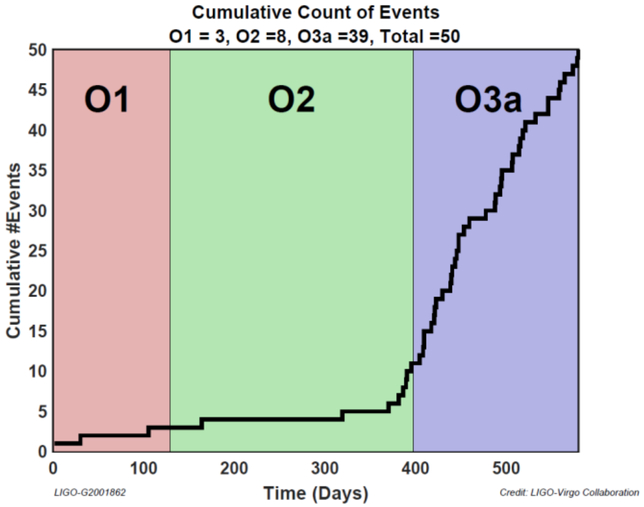
Gravitational Waves Detections

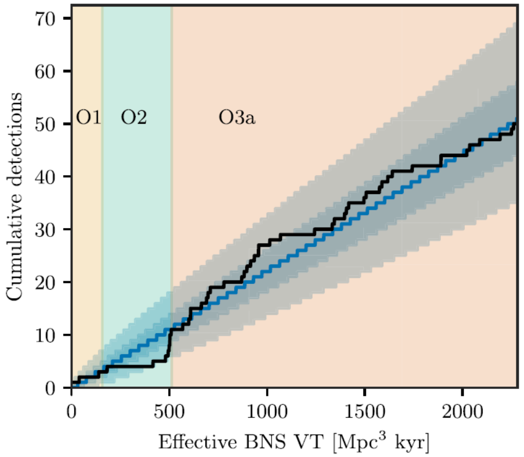
Brief history of GW Observations

1915 General Relativity

1960s Webber's bar detector

1970s pulsar-neutron star discovery





PRX11, 021053(2021)

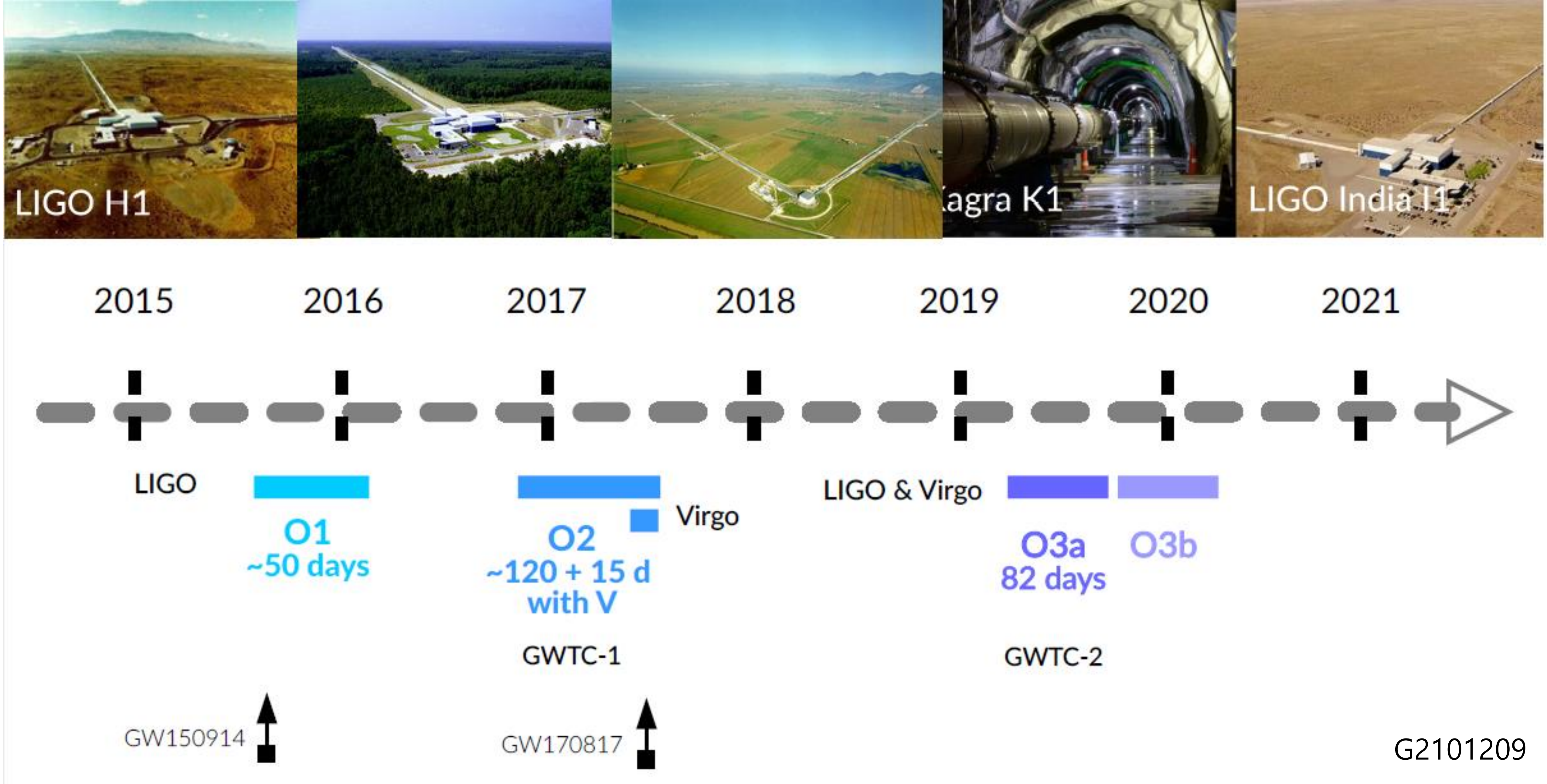
O1: 2015/09/12~2016/01/19
3 BBH direct detection(GW150914, GW151012, GW151226)

O2: 2016/11/30~2017/08/25
7 BBH(GW170104, GW170608, GW170729, GW170809, GW170814, GW170818, GW170823) and
1 BNS(GW170817)

O3a : 2019/04/01~2019/09/30
O3b : 2019/11/01~2020/03/27(COVID-19 shutdown)

Around 40 events

[List of gravitational wave observations - Wikipedia]


LIGO H1
Kagra K1
LIGO India I1
2015
2016
2017
2018
2019
2020
2021
LIGO
O1
~50 days
O2
~120 + 15 d
with V
Virgo
LIGO & Virgo
O3a
82 days
O3b
GWTC-1
GWTC-2
GW150914
GW170817

G2101209

GWTC-2

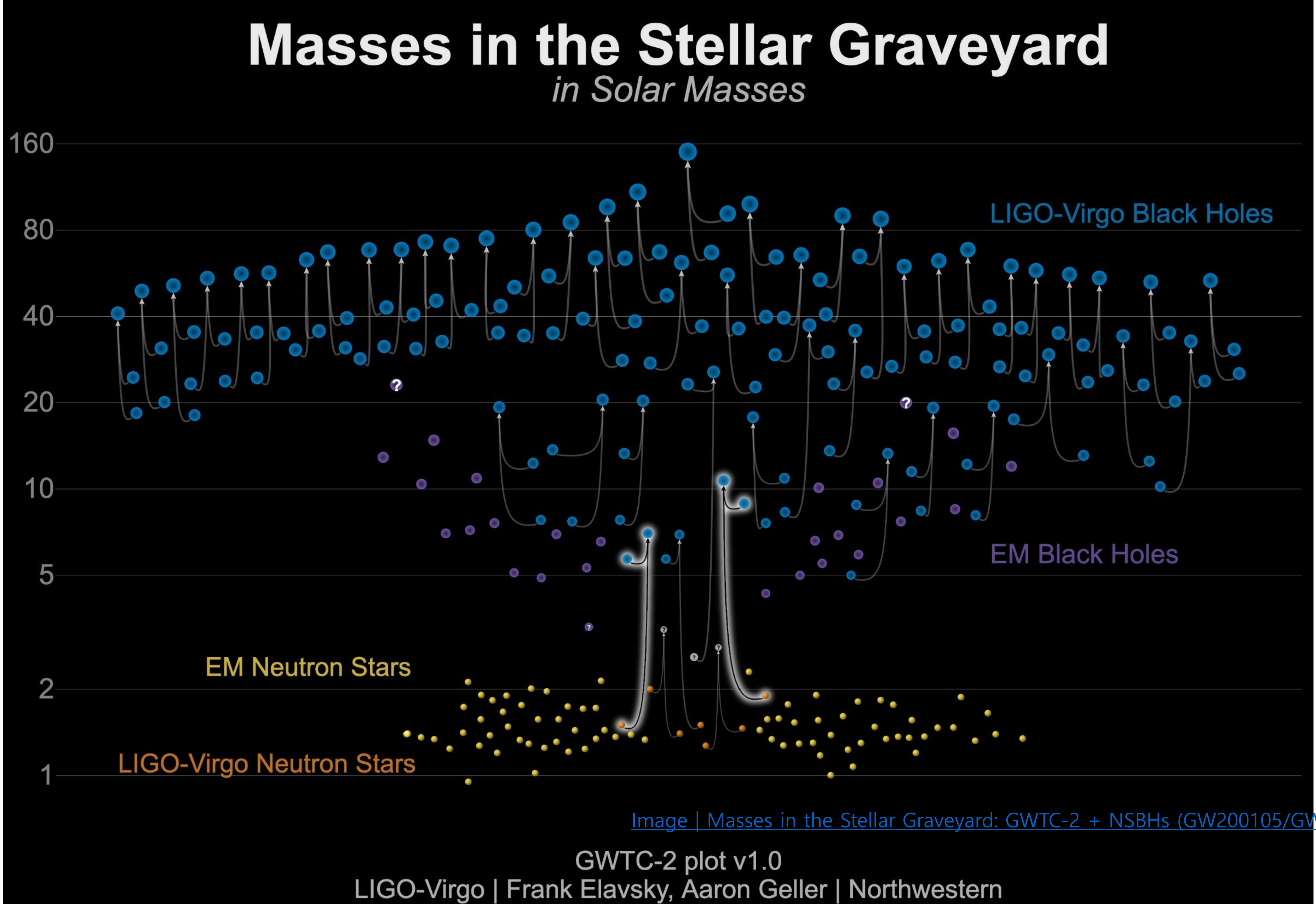
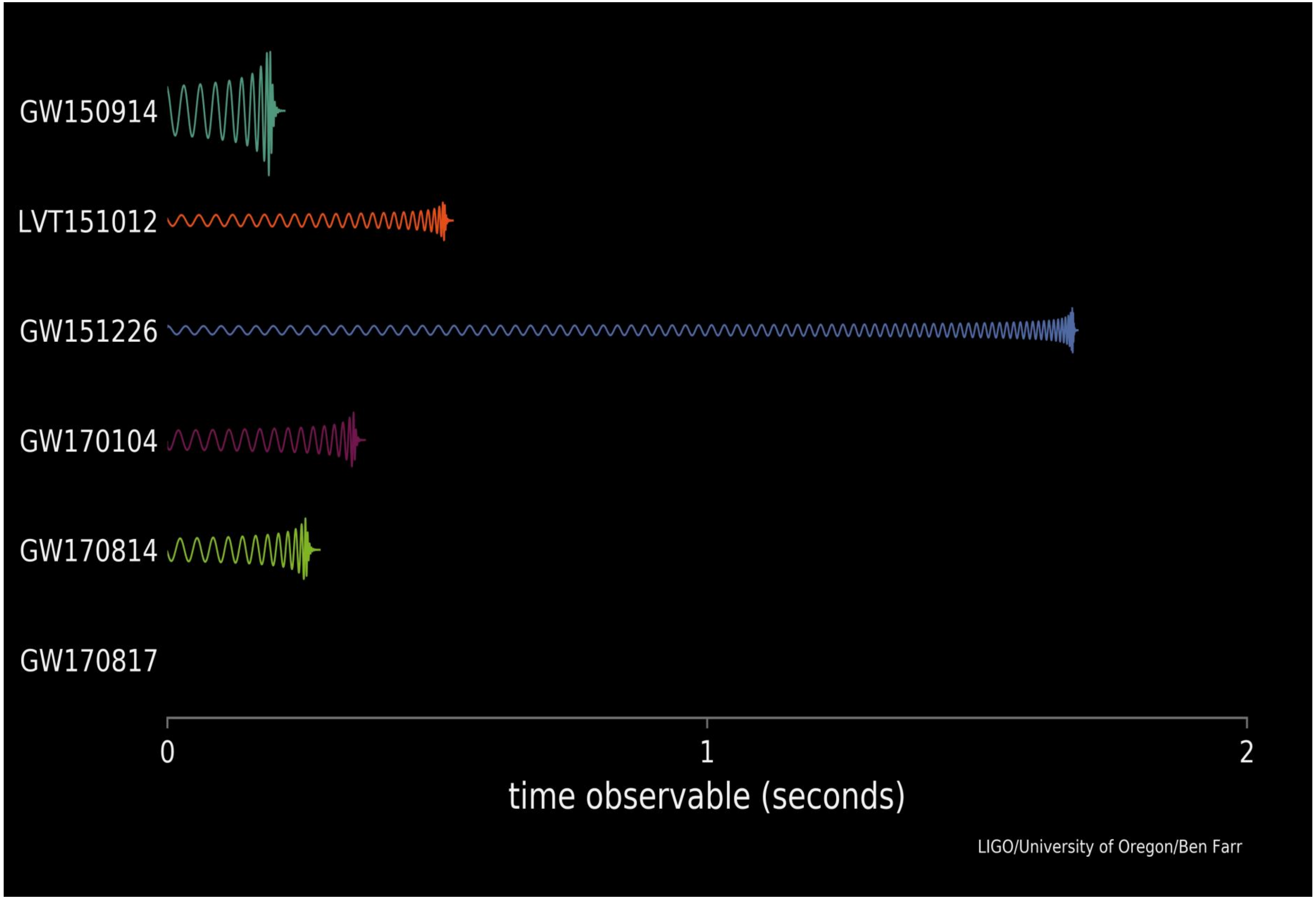


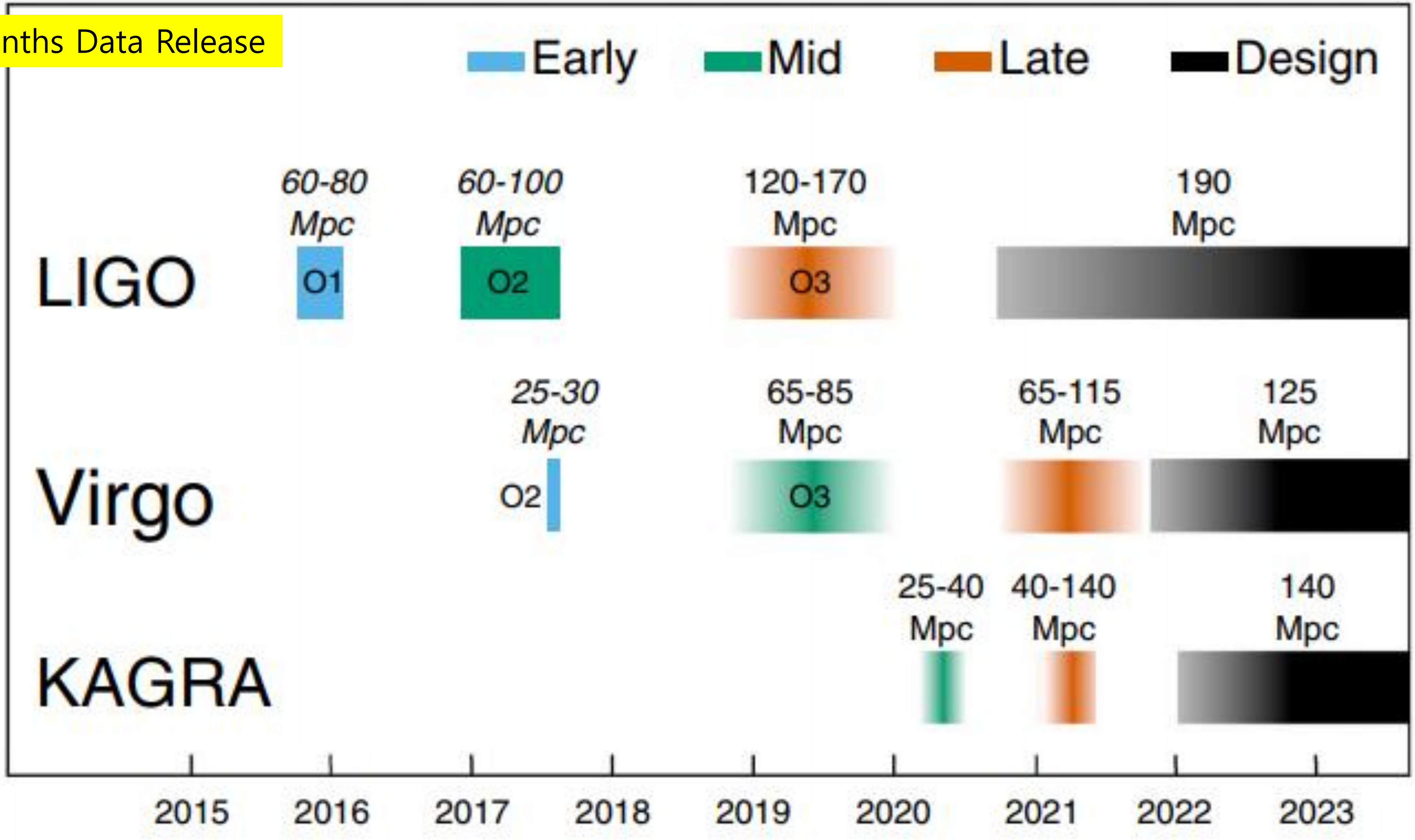
Image | Masses in the Stellar Graveyard: GWTC-2 + NSBHs (GW200105/GW200115) | LIGO Lab | Caltech

GW signal

[GWOSC (gw-openscience.org)](gw-openscience.org)

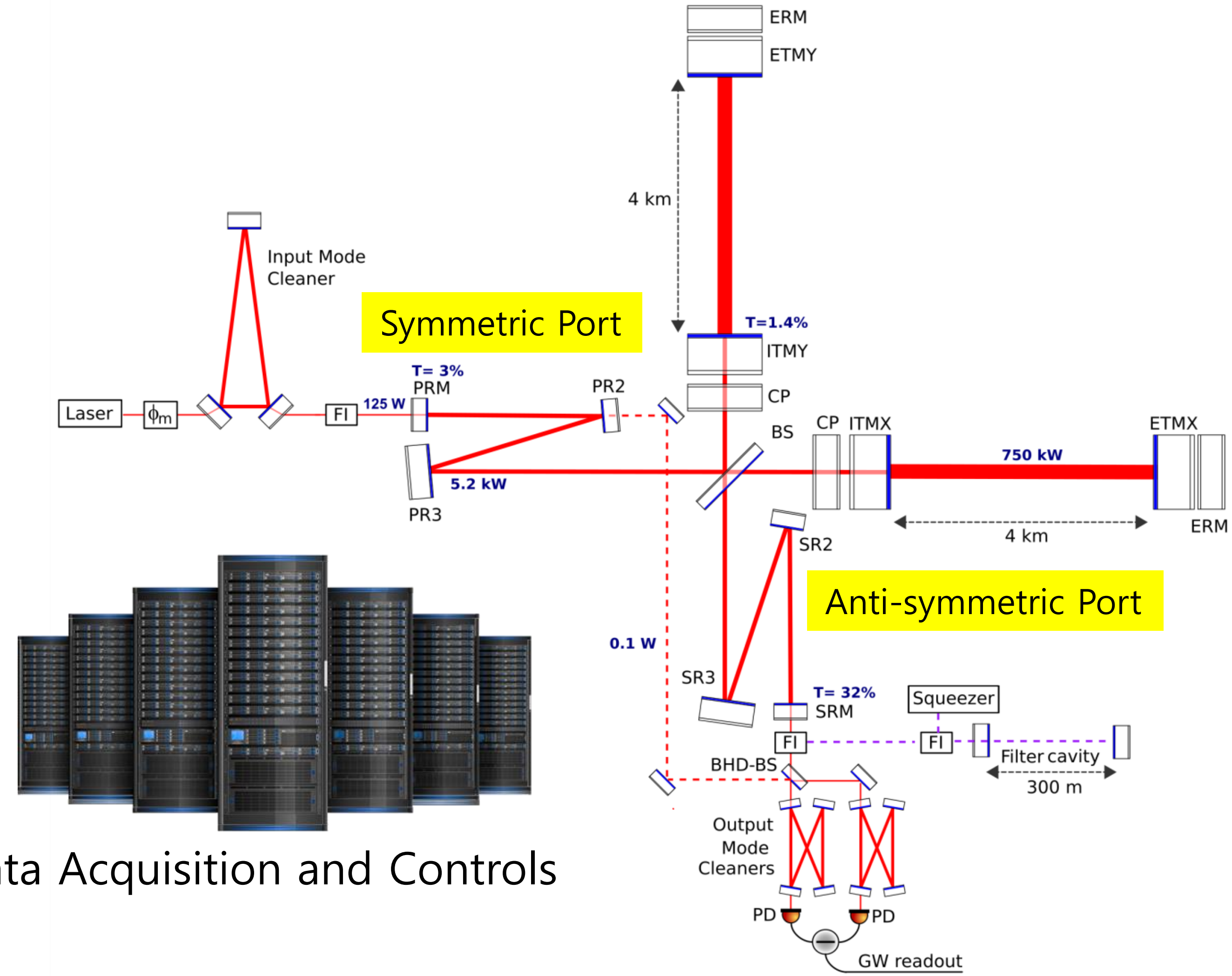


18 months Data Release



How Data Obtained

Detector operation



Data Acquisition and Controls

- Sampling rate
  - 16384Hz:LIGO
  - 20kHz:Virgo
  - 16384Hz:KAGRA
- Valid range
  - 10Hz~5kHz : LIGO
  - 10Hz~8kHz : Virgo
  - 10Hz~5kHz : KAGRA

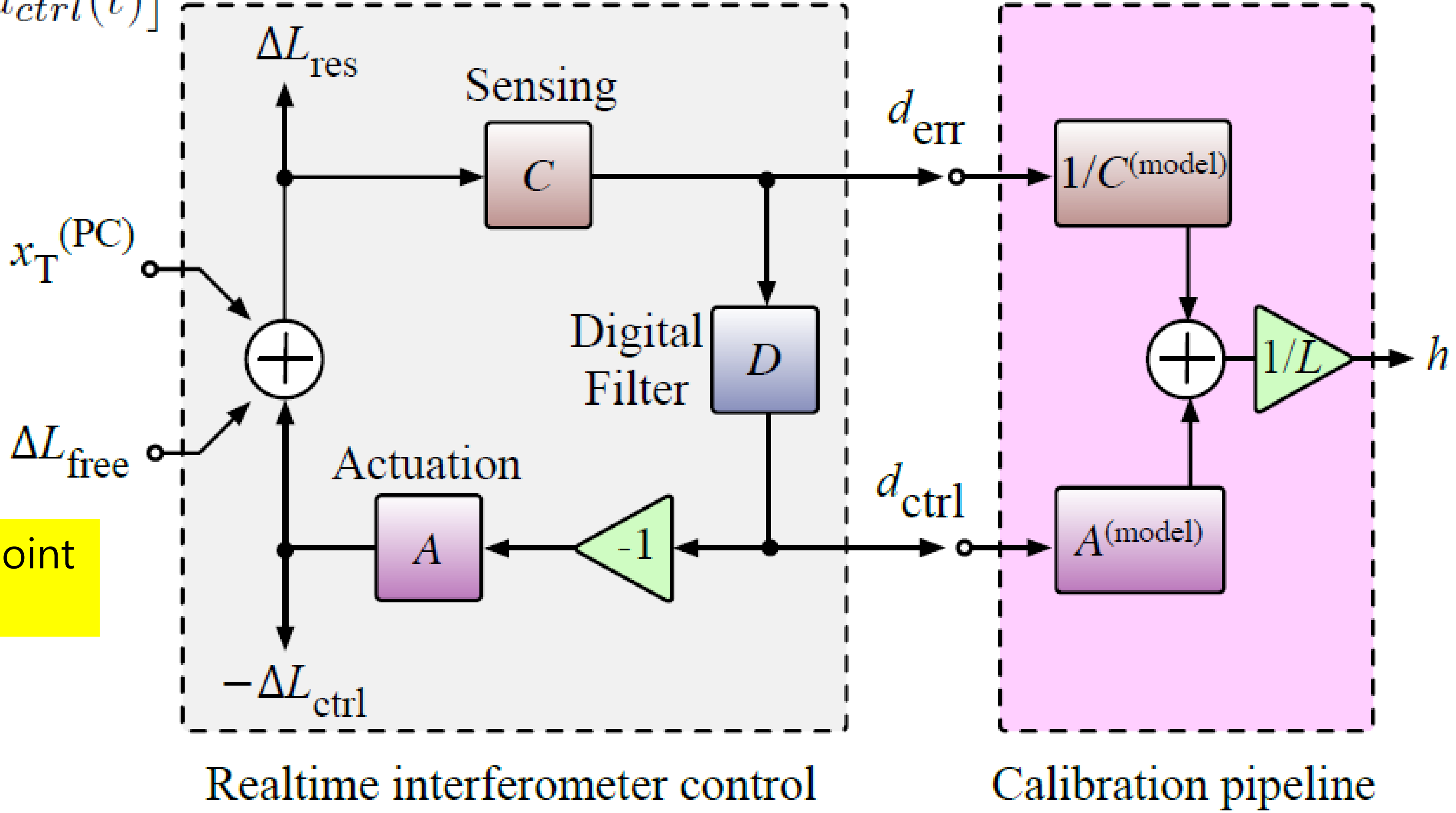
Generation of h(t)

$$d_{err} = \mathcal{C}(\Delta L_{free} + x_T - \mathcal{A} d_{ctrl})$$
$$\Delta L_{free} = \mathcal{C}^{-1} d_{err} + \mathcal{A} d_{ctrl} - x_T$$

$$h(t) = \frac{1}{L}\left[\mathcal{C}^{-1} * d_{err}(t) + \mathcal{A} * d_{ctrl}(t)\right]$$

$$h(t) = \frac{\Delta L_{free}}{L}$$

Keep interferometer at “operation” point
No difference appear dark point



Data Analysis

Data Analysis

- GW signal is very weak $\sim 10^{-22} - 10^{-24}$
- Matched filter for known waveform
- Detector characterization

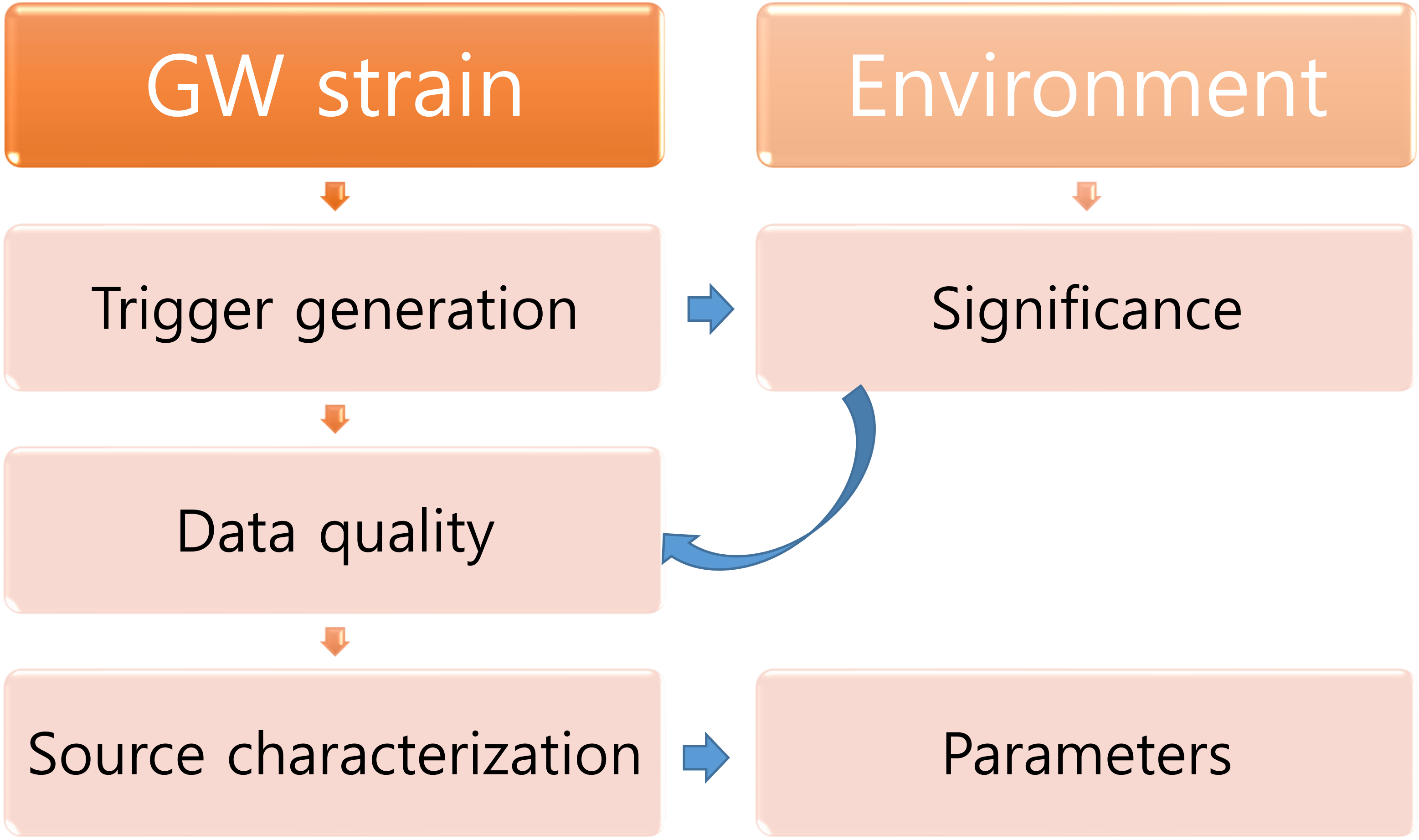
Data Analysis

Detector Characterization
- Noise identification
- Detector performance

Search
- Matched filter
- Masses, Distance, Sky location

Parameter Estimation
- Better accuracy
- Distribution

Data Analysis Pipeline



Credit V. Raymond

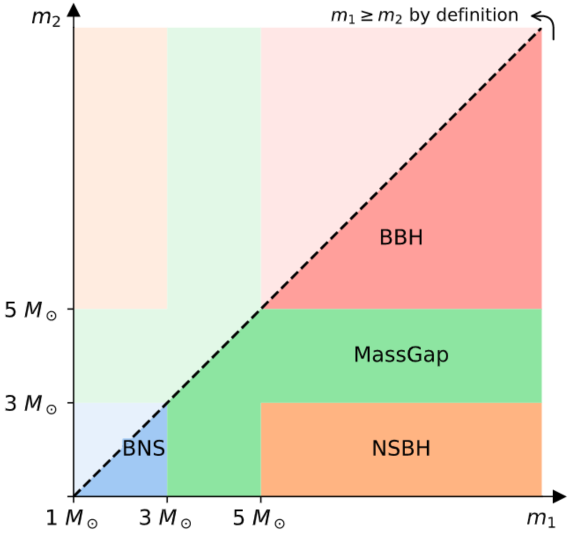
Data flow

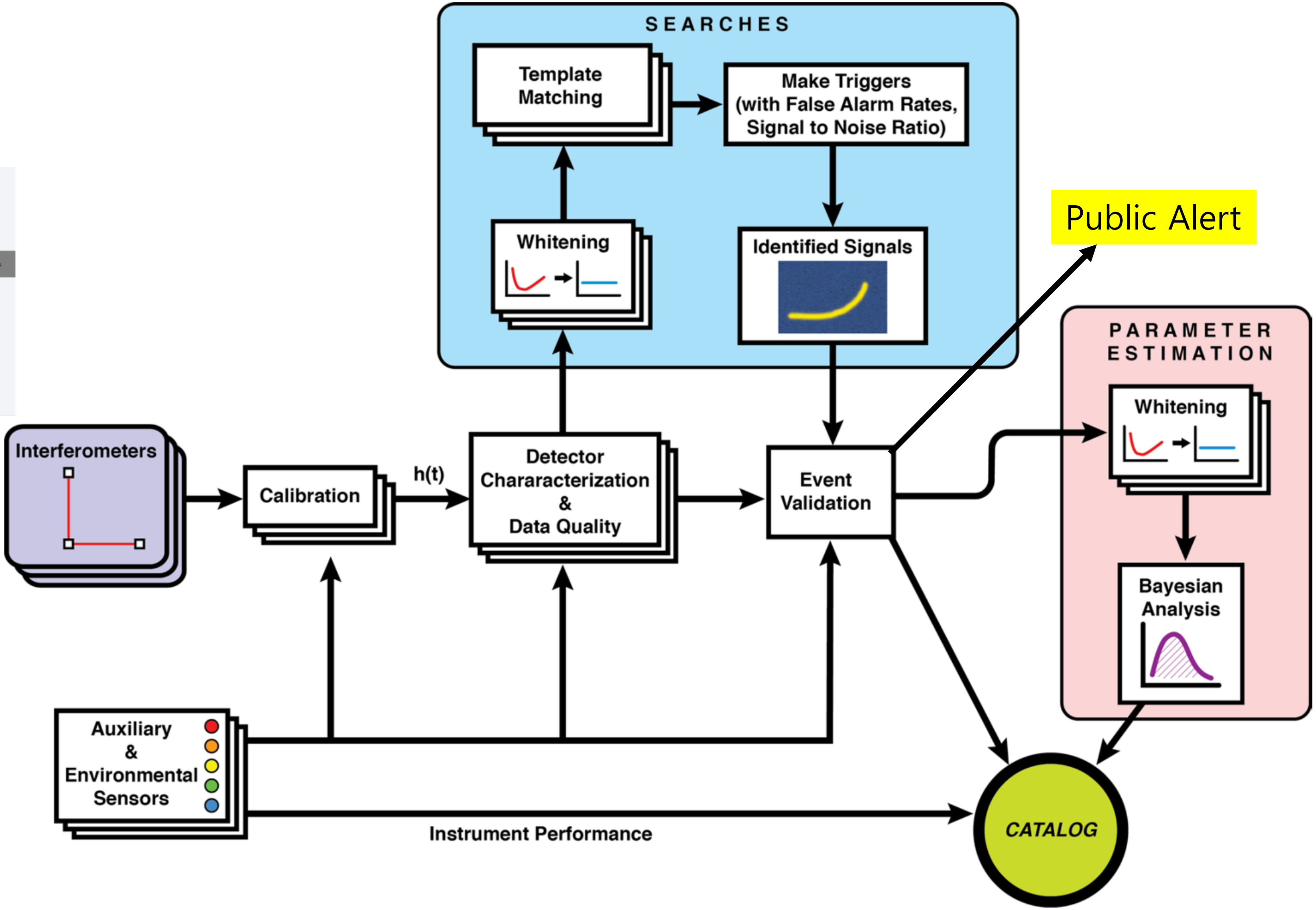
LIGO/Virgo O3 Public Alerts

Detection candidates: 56

SORT: EVENT ID (A-Z)

Event ID	Possible Source (Probability)	UTC	GCN	Location	FAR	Comments
S200316bj	MassGap (>99%)	March 16, 2020 21:57:56 UTC	GCN Circulars Notices | VOE		1 per 446.44 years	
S200311bg	BBH (>99%)	March 11, 2020 11:58:53 UTC	GCN Circulars Notices | VOE		1 per 3.5448e+17 years	





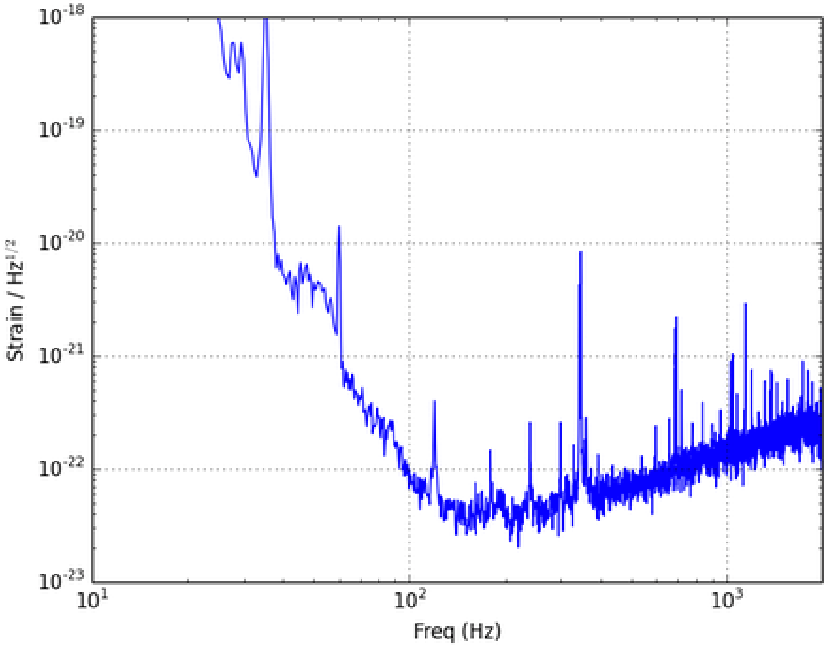
Gravitational Wave Open Science Center gw-openscience.org

G2101351

- Data Downloads
- Online courses
- Tutorials
- Documentation

Amplitude Spectral Density (ASD)

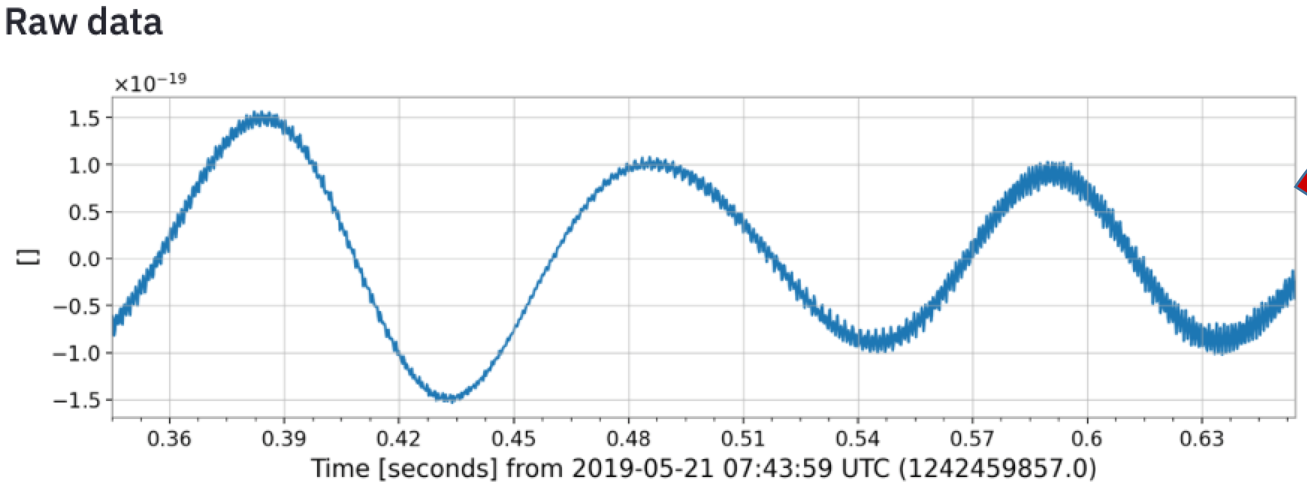
The ASD can be obtained by taking the square root of the PSD. This is done to give units that can be more easily compared with the time domain data or FFT. More information about the LIGO ASD can be seen on the Instrumental Lines page.



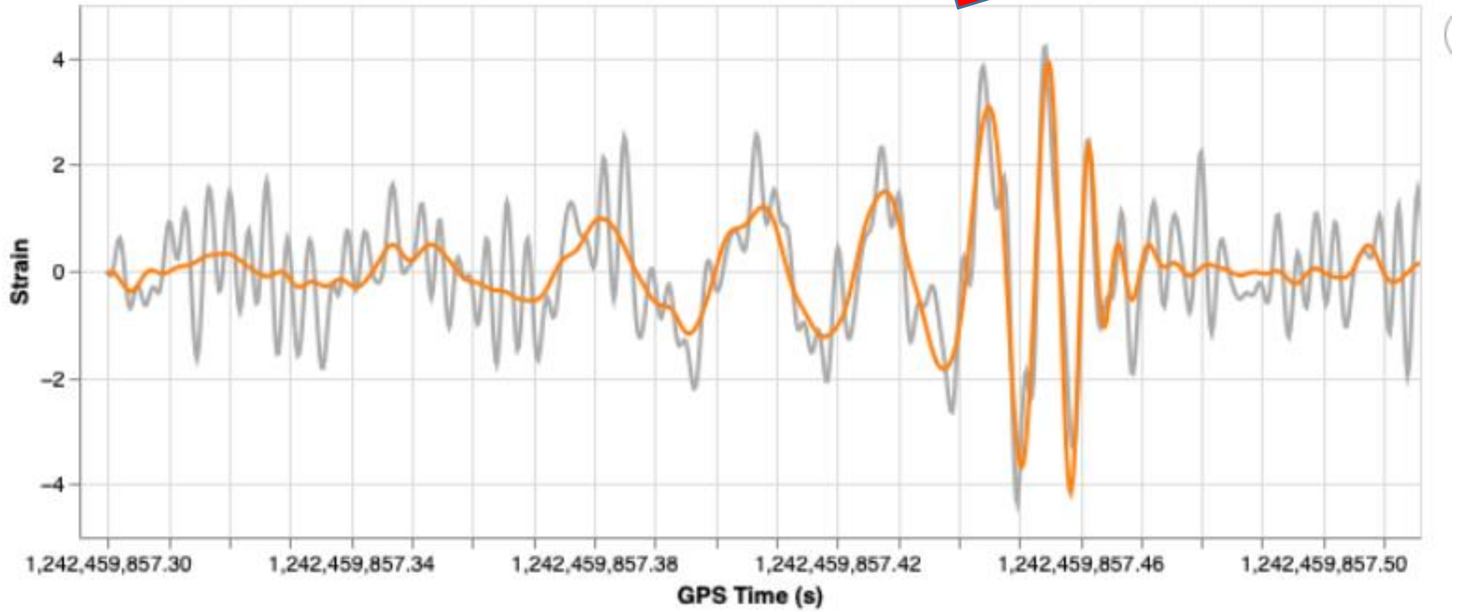
Show the code

```
#--------------------------
# Plot the ASD
#--------------------------------
Pxx, freqs = mlab.psd(strain_seg, Fs=fs, NFFT=2*fs)
plt.loglog(freqs, np.sqrt(Pxx))
plt.axis([10, 2000, 1e-23, 1e-18])
plt.grid('on')
plt.xlabel('Freq (Hz)')
plt.ylabel('Strain / Hz$^{1/2}$')
```

Finding signals in noisy data

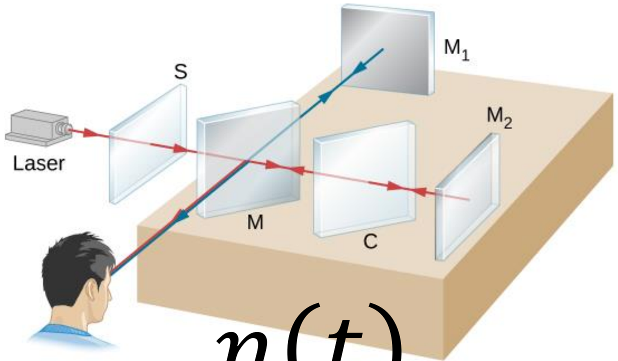


Optimal Filter



Linear Optimal Filter

$h(t)$



$n(t)$

$$d(t) = h(t) + n(t)$$

$$\varepsilon(t) = \mathcal{H}\big(d(t)\big) - h(t)$$

$$\text{minimize } \overline{\varepsilon^2}$$

$$K(f) \propto \frac{h(f)}{S_n(f)}$$

Matched Filter

- Correlate data with expected signal (Here, plotting absolute value) G1400343

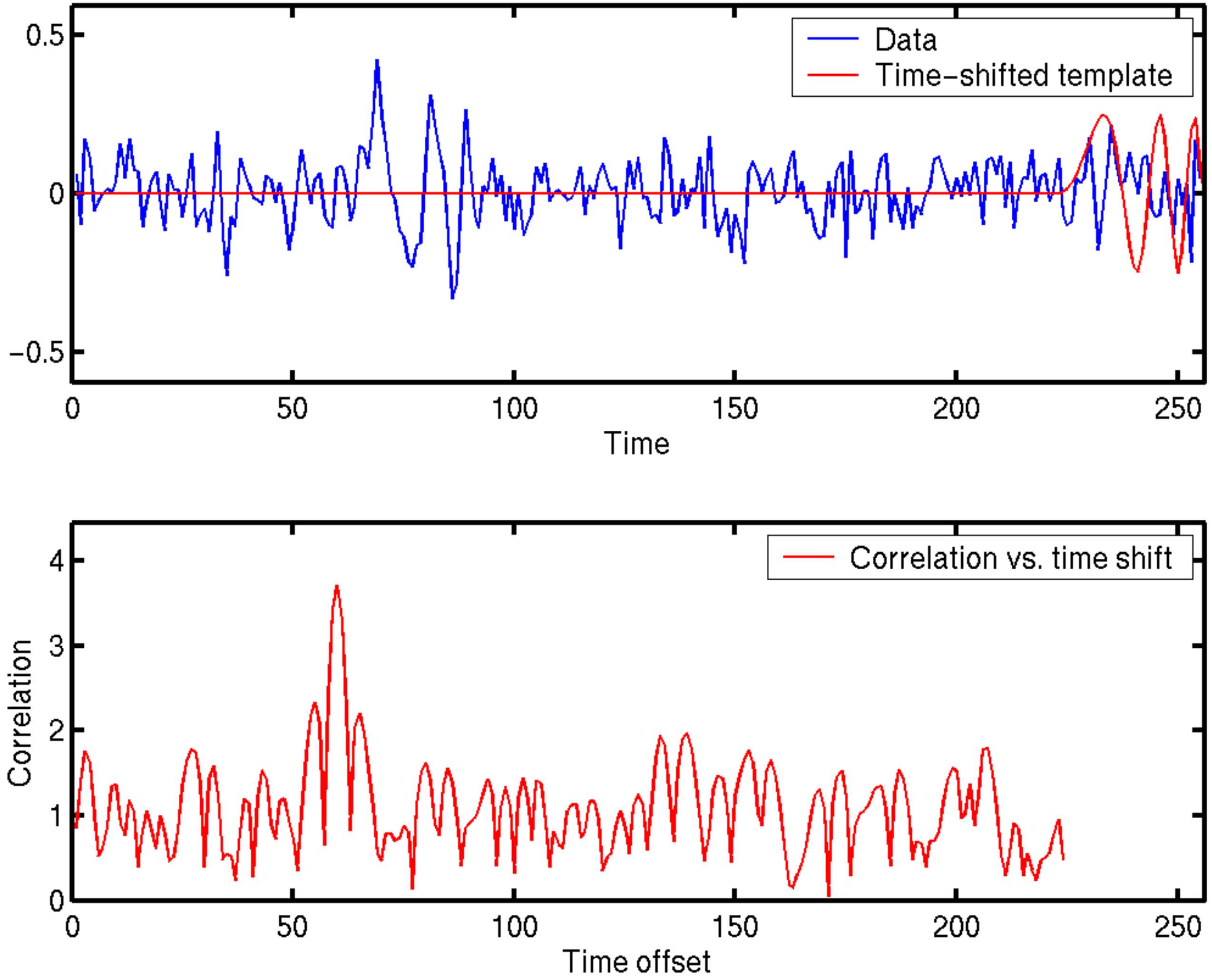
$$C(t) = \int_{-\infty}^{\infty} dt' \; s(t') \;\; h(t' - t)$$

Time offset

Data

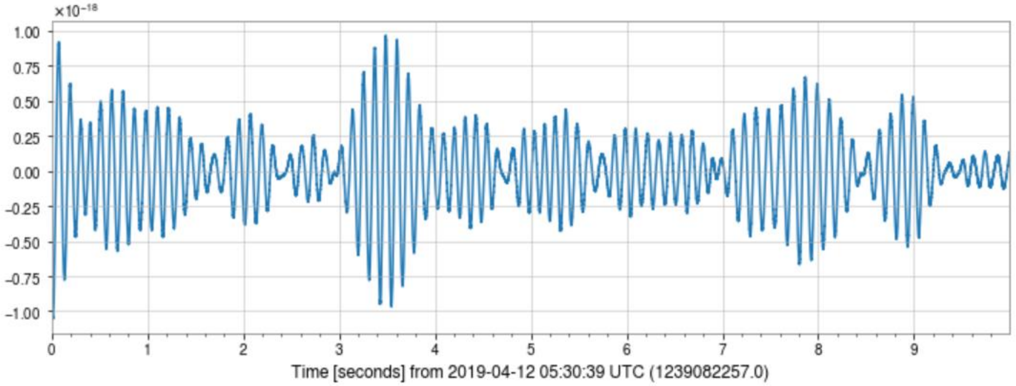
Template with time offset

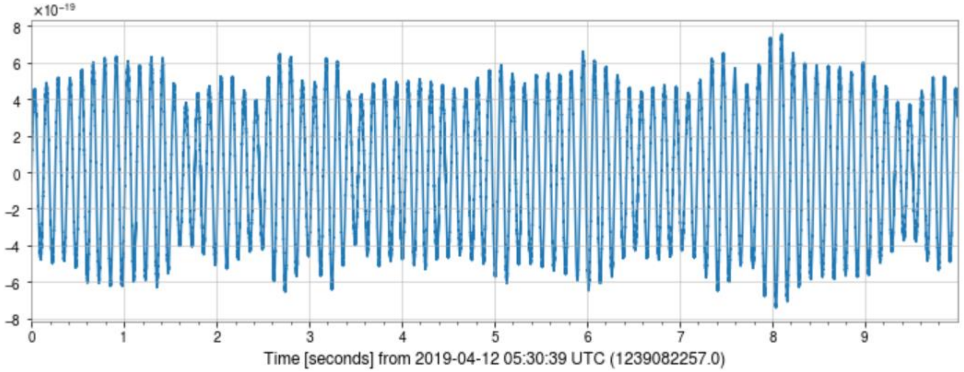
$$C(t) = 4 \int_0^{\infty} \frac{\tilde{s}(f) \;\; \tilde{h}^*(f)}{S_n(f)} e^{2\pi i f t} df$$

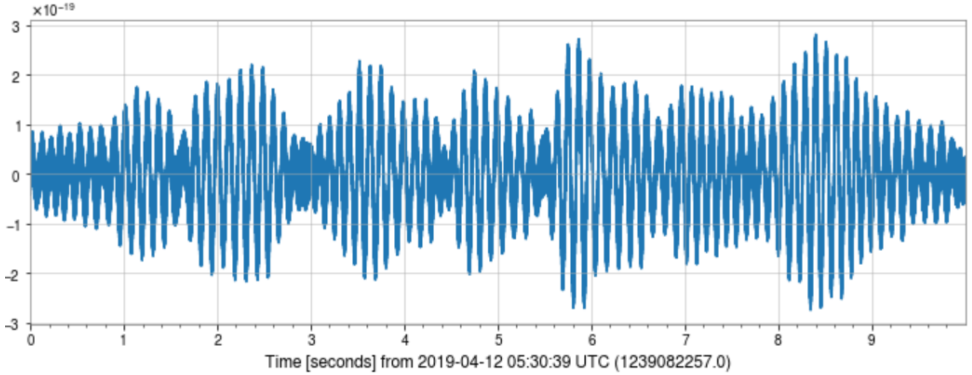


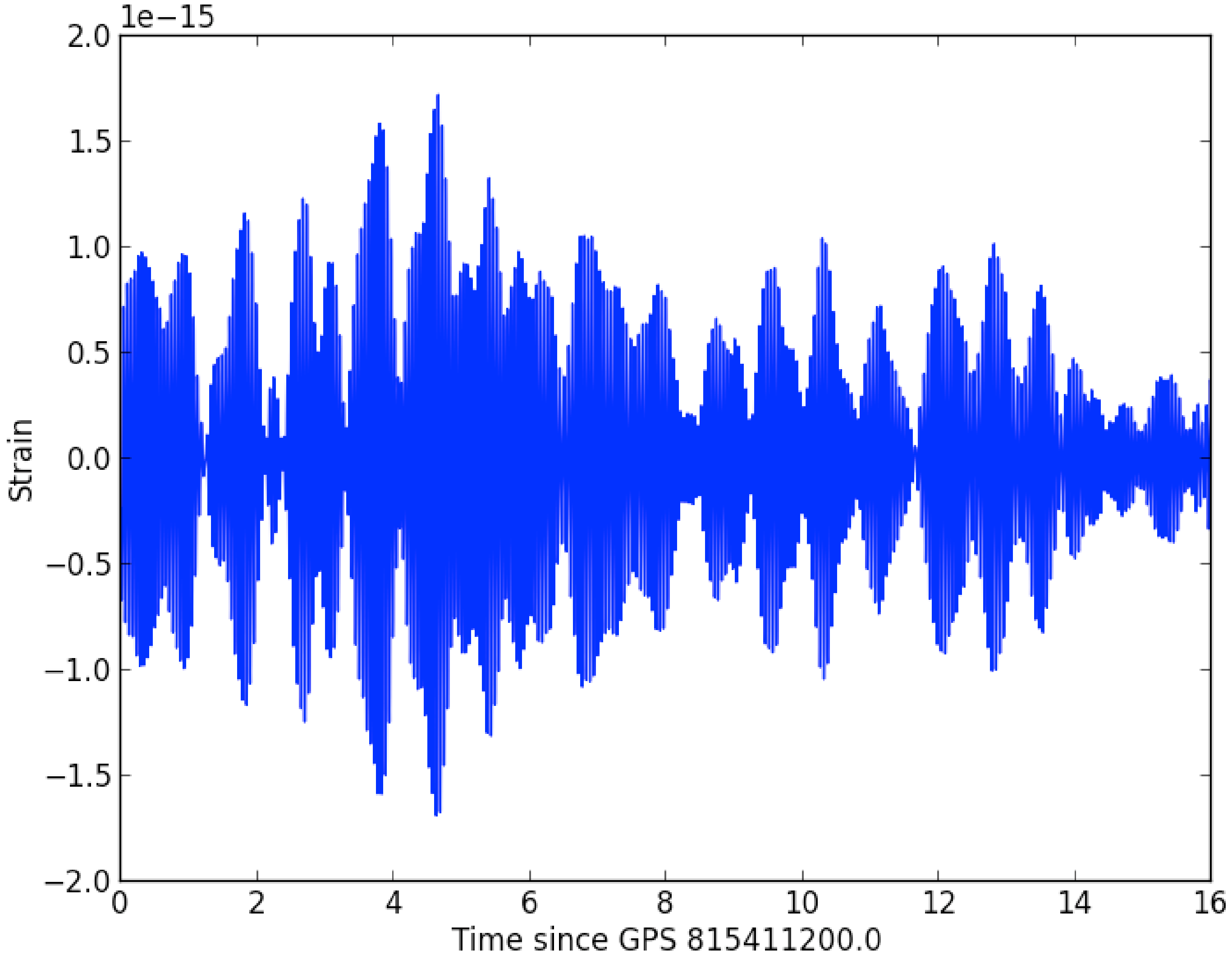
Time Series Data
Nyquist Frequency

Some LIGO Data





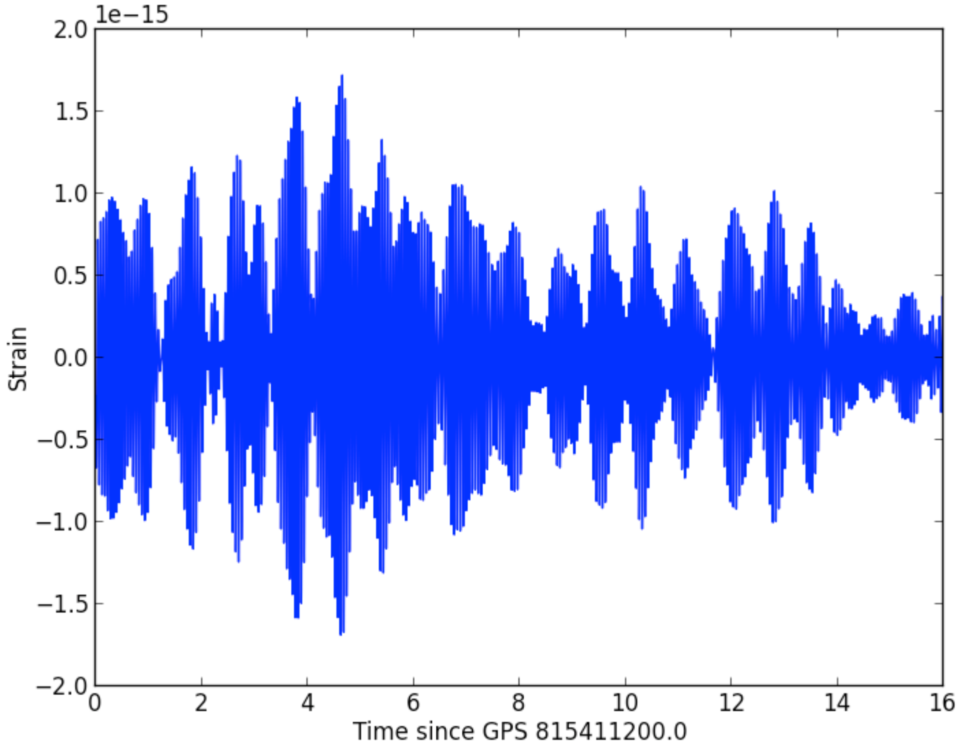




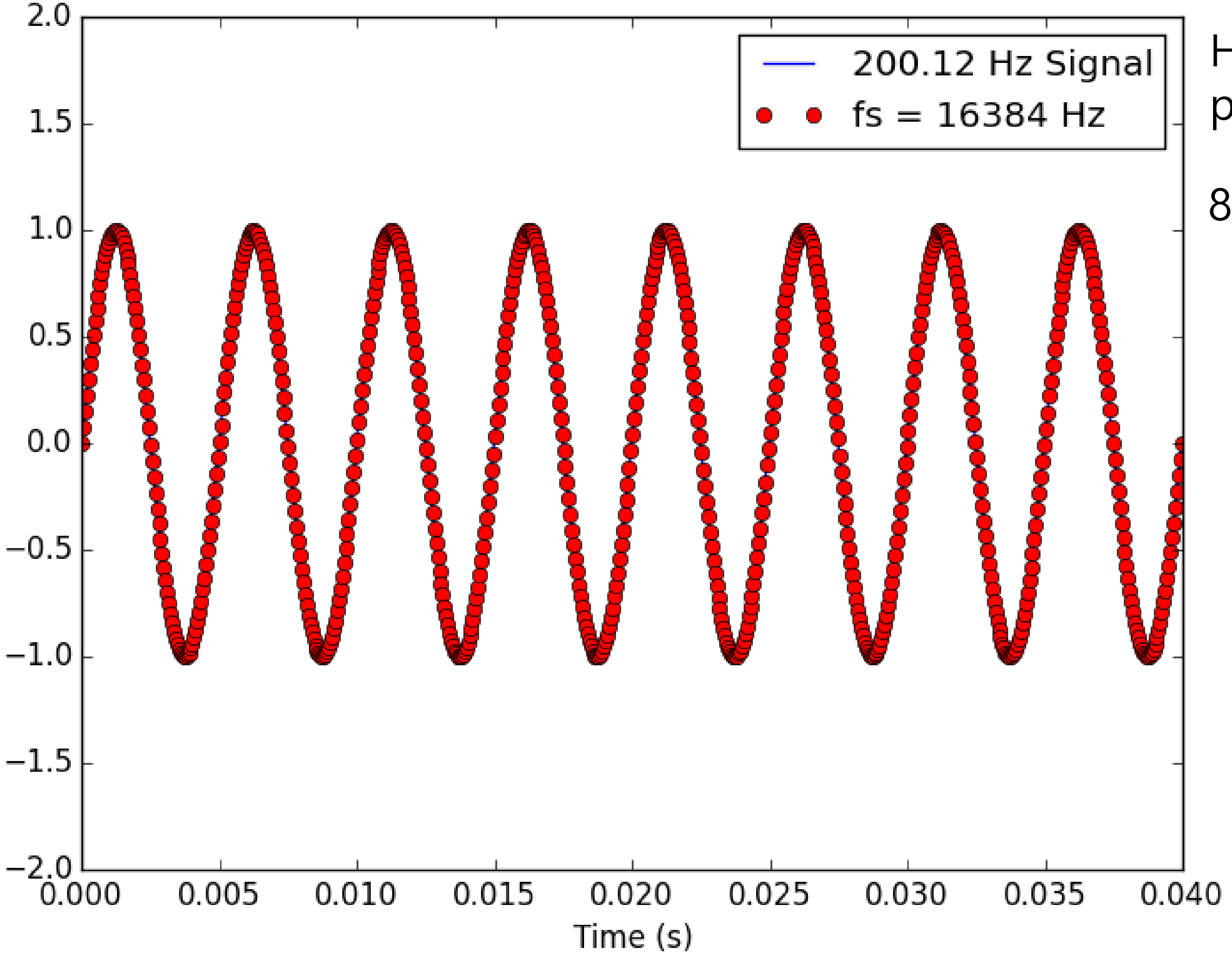
Odw4, GW190412

LIGO Data

- Discretely sampled time-series data
- Sampling rate (fs)
- h(t) – calibrated strain
  - ALSO: hundreds of “auxiliary” channels
- Recorded at 16384 Hz sample rate
- ~300 MB per hour
- Stored in .gwf “frame” files
  - Also HDF5(Hierarchical Data Format version 5)



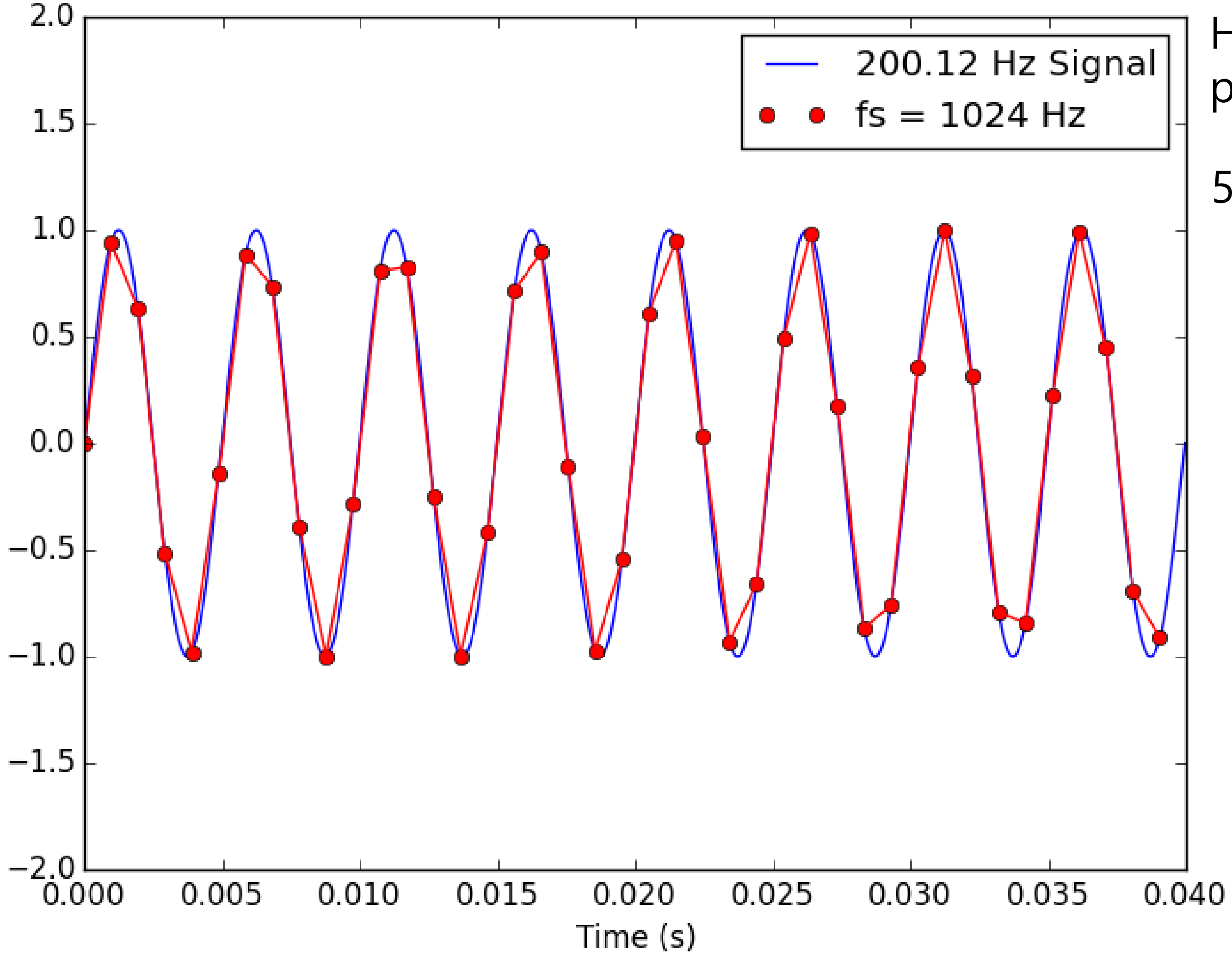
Discrete Time Samples



How many samples per cycle?

81.87 times

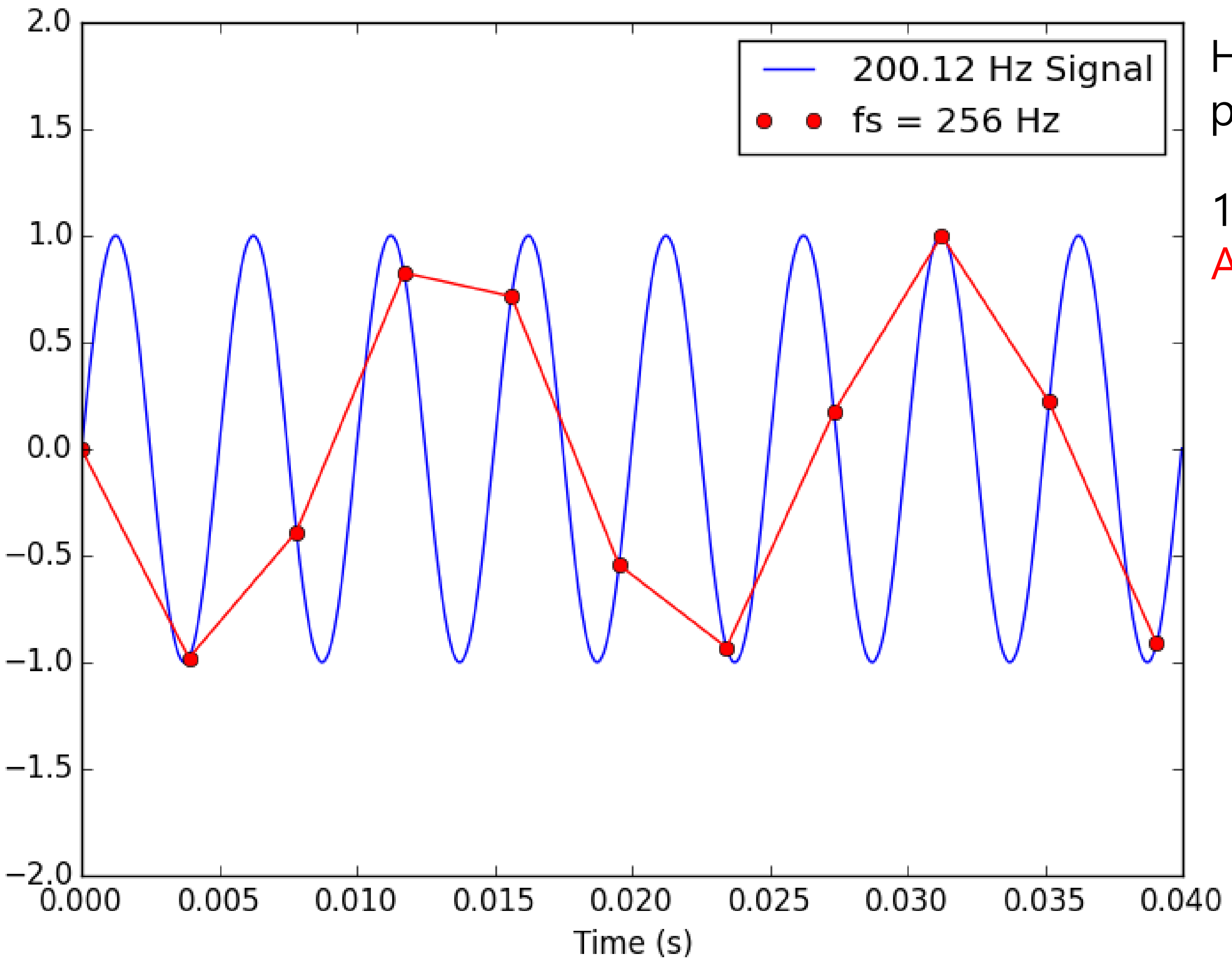
Discrete Time Samples



How many samples per cycle?

5.12 times

Discrete Time Samples



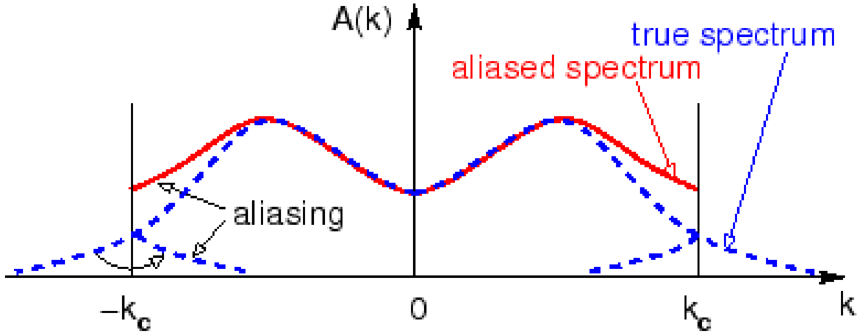
How many samples per cycle?

1.28 times

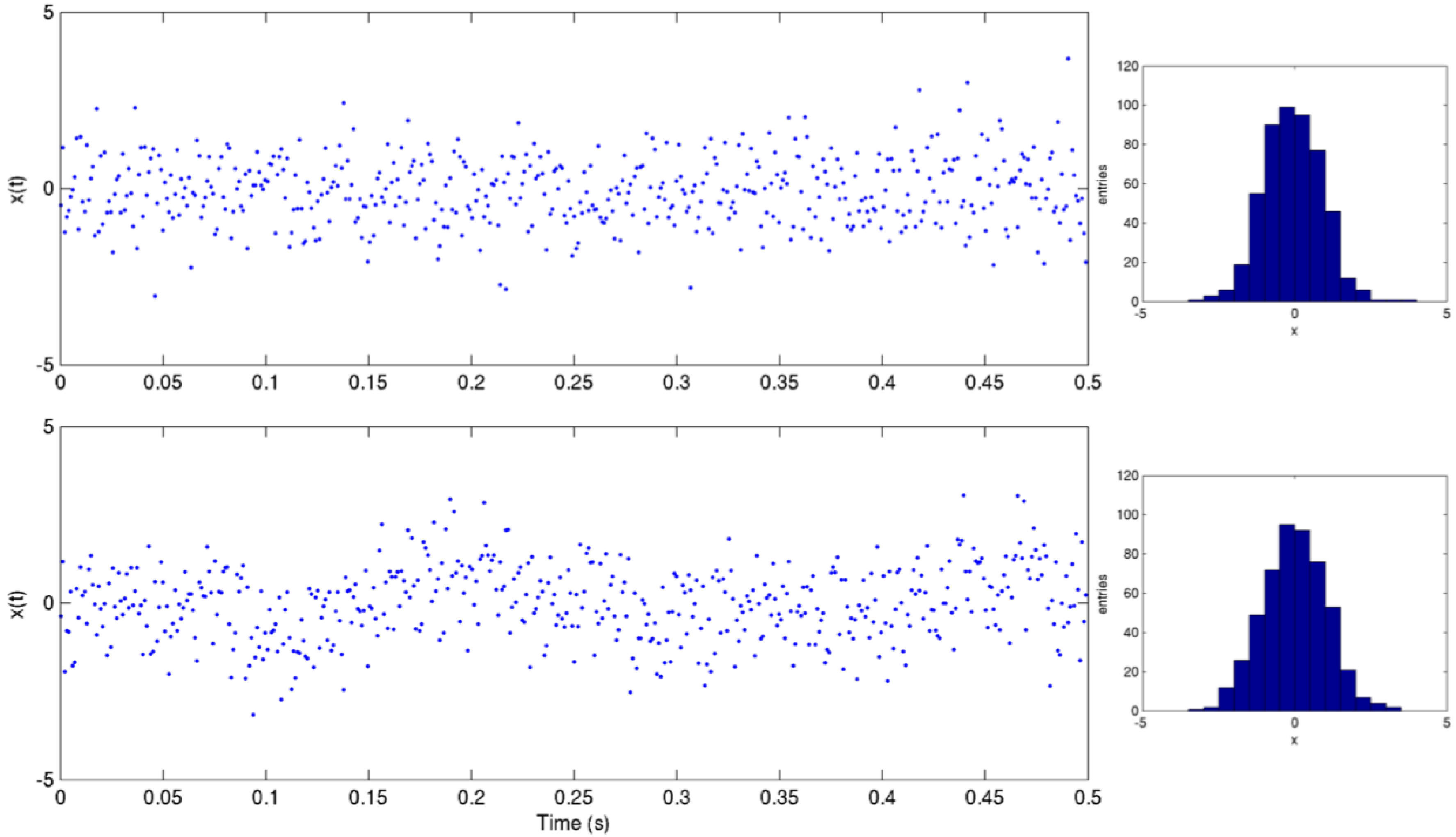
Aliasing occurs

Nyquist Frequency

- Nyquist Frequency = f_s / 2
  - Discretely sampled data with sampling rate fs can represent a continuous signal which only has frequency content below the Nyquist frequency
- Data can only contain frequency content below the *Nyquist frequency*
- Higher frequency signals will be lost or "aliased" to lower frequencies



Noise is random, but its properties can be characterized



Slide from Peter Shawhan

Possible properties of noise

Stationary : statistical properties are independent of time

Ergodic process: time averages are equivalent to ensemble averages

Gaussian : A random variable follows Gaussian distribution

For a single random variable, $$p(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left[-\frac{1}{2}\frac{(x-\mu_x)^2}{\sigma_x^2}\right]$$

More generally, a *set* of random variables (e.g. a time series) is Gaussian if the joint probability distribution is governed by a covariance matrix

$$C_{xij} := \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle$$

such that

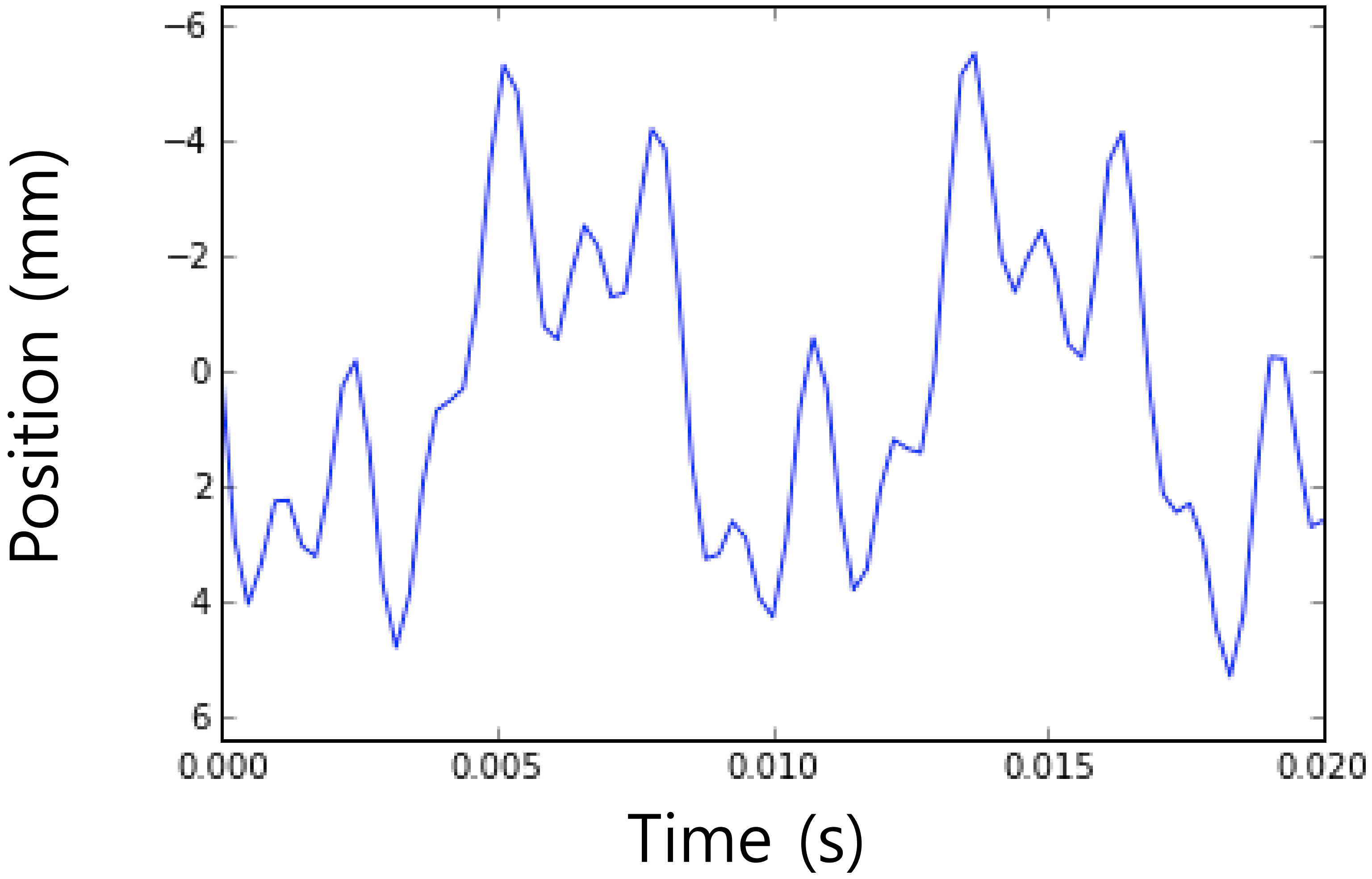
$$p(x_1, x_2, \ldots, x_N) = \frac{1}{(2\pi)^{N/2}\sqrt{\det C_x}} \exp\left[-\frac{1}{2}\sum_{i,j=0}^{N-1} C_{xij}^{-1}(x_i - \mu_{xi})(x_j - \mu_{xj})\right]$$

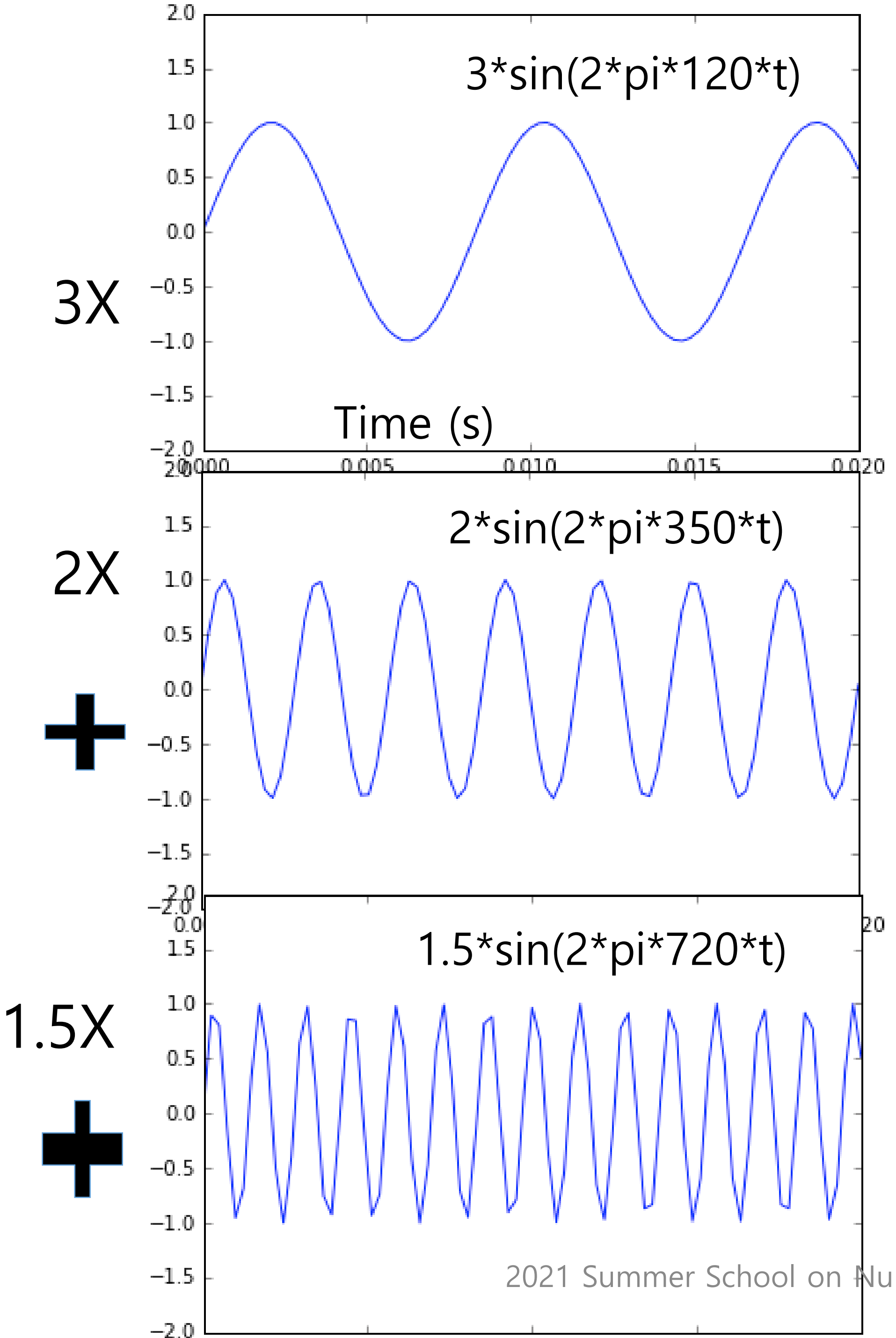
White : Signal power is uniformly distributed over frequency

$\Rightarrow$ Data samples are uncorrelated

Work in Frequency Domain

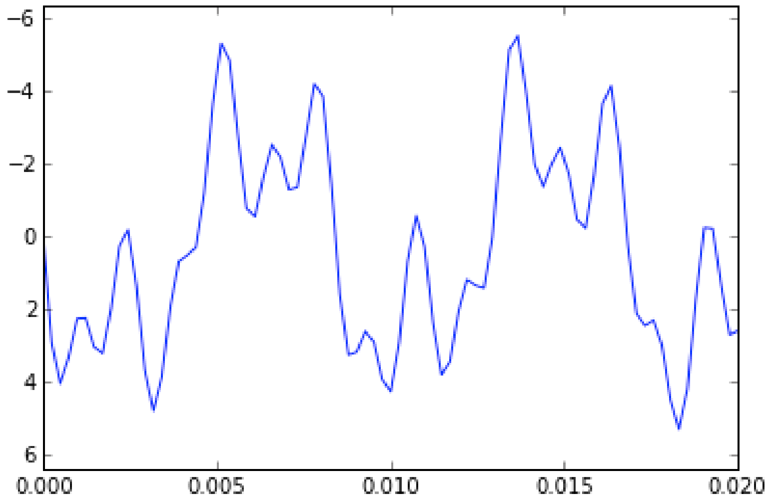
How would you describe this function?





=

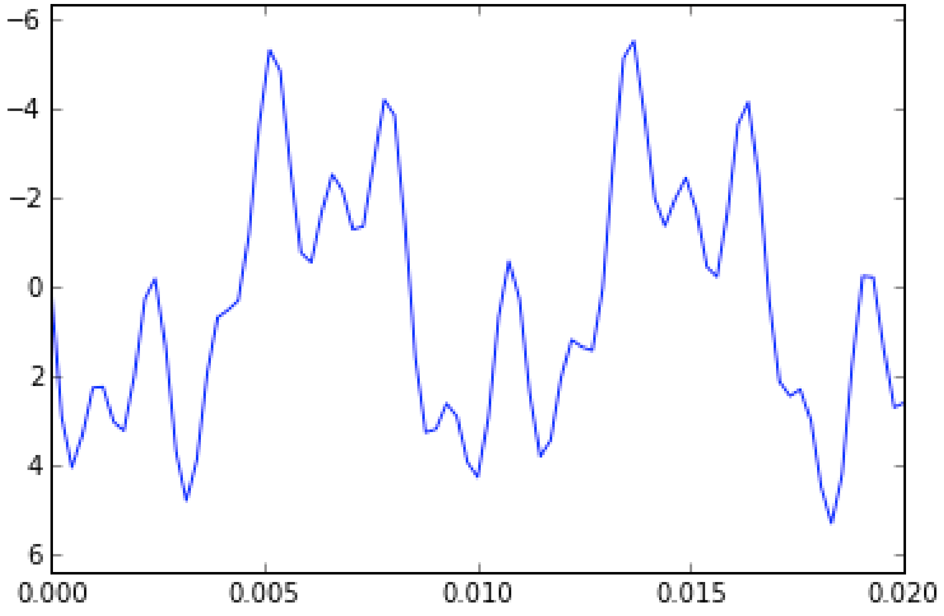
Noise in the time domain



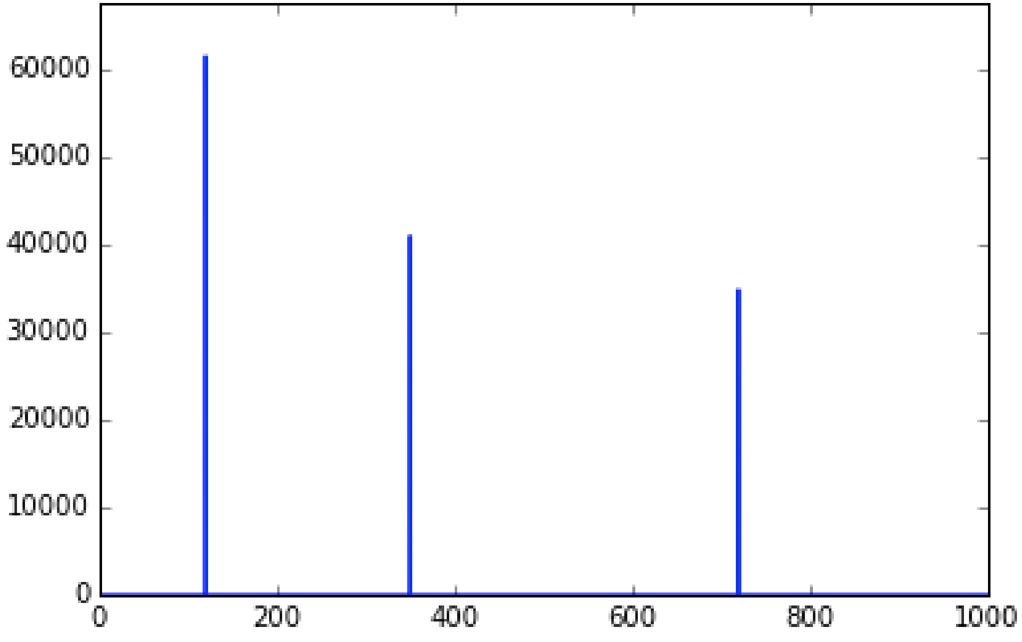
Amplitude -> Loudness -> “Power”

Frequency -> Pitch

Time Domain



Frequency Domain



h(t) – Position as a function of time

=

H(f) – Amplitude and phase 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

Key Concept

Any function can be represented as a sum of sine waves.

$$h(t) = A_1 \sin(\omega_1 t + \varphi_1) \quad + \quad A_2 \sin(\omega_2 t + \varphi_2) \ldots$$

In the "frequency domain", we are plotting the amplitudes (or phases) of each of these sine waves.

$$|H(\omega_n)| = A_n$$

Fourier Transform is used to represent data in the frequency domain

Fourier transform

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

$$\Rightarrow \qquad x(t) = \int_{-\infty}^{\infty} df\, \widetilde{x}(f) e^{i2\pi f t}$$

$|\widetilde{x}(f)|^2$ **can be interpreted as energy spectral density**

Efficient way to calculate complete discrete Fourier Transform:
Fast Fourier Transform (FFT)

Power Spectral Density

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

$|\tilde{x}(f)|^2$ can be interpreted as energy spectral density

When noise (or signal) has infinite extent in time domain, can still define the power spectral density (PSD)

$$\lim_{T\to\infty} \frac{1}{T} \left|\tilde{x}_T(f)\right|^2 \qquad \langle n(f) n^*(f') \rangle = \delta(f - f') S_n(f)$$

Watch out for one-sided vs. two-sided PSDs

Slide from P. Shawhan

Estimating the PSD

Simplest approach: FFT the data, calculate square of magnitude of each frequency component – this is a periodogram

- For stationary noise, one can show that the frequency components are statistically independent

This estimate is unbiased (has the correct mean), but has a large variance – so average several periodograms

- Alternately, smooth periodogram; give up frequency resolution either way

Generally apply a “window” to the data to avoid spectral leakage

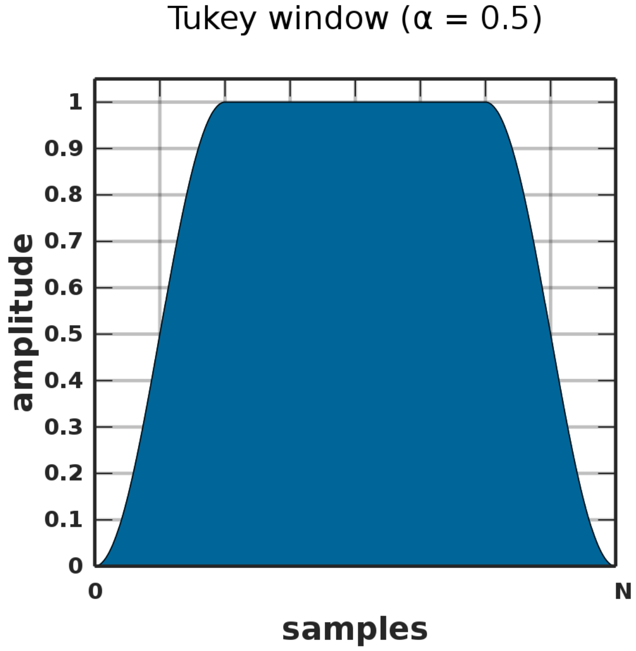
- Leakage arises from the assumption that the data is periodic!
- Tapered window forces data to go to zero at ends of time interval

Welch’s method of estimating a PSD averages periodograms calculated from windowed data

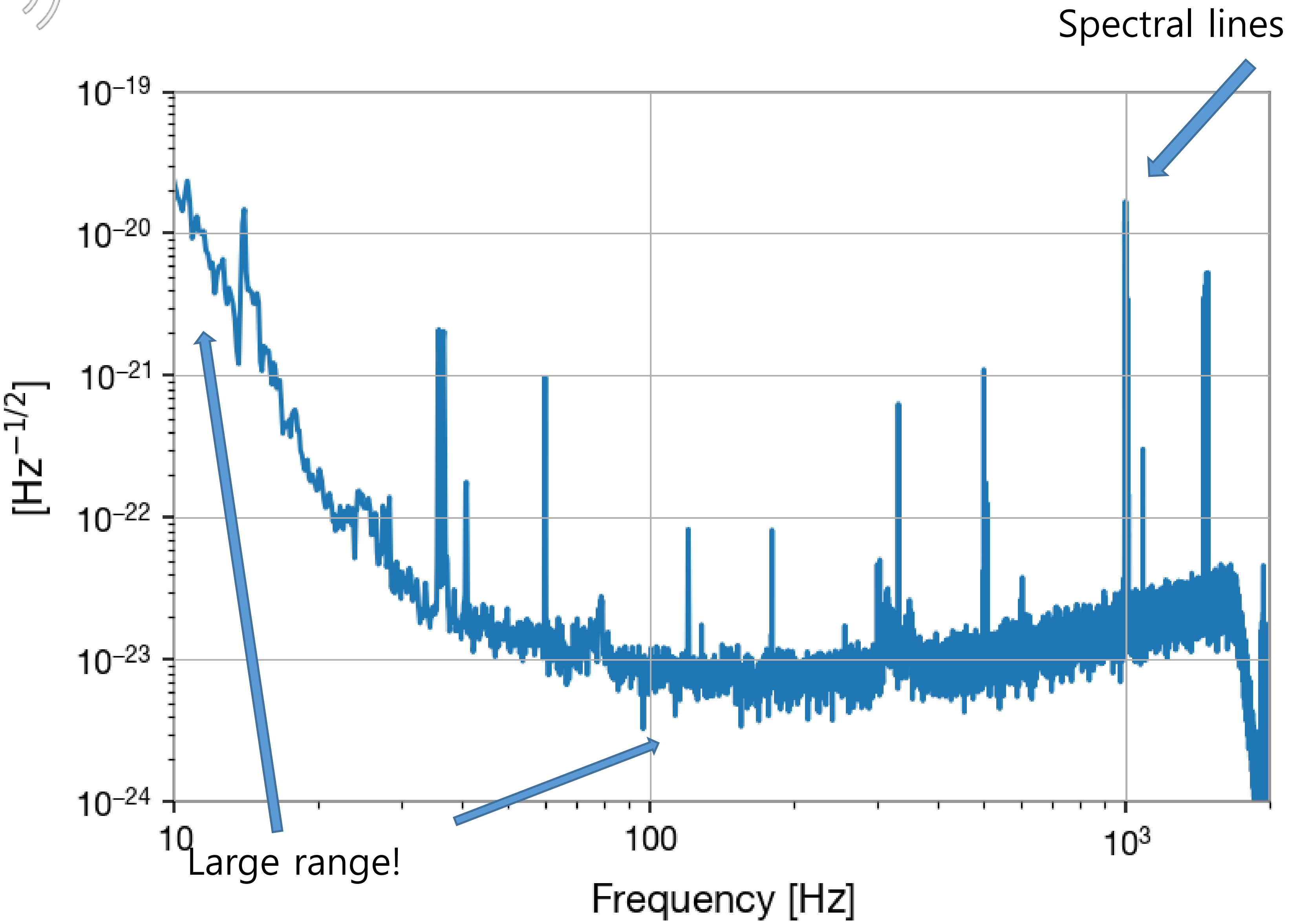
Slide from P. Shawhan

PSD Estimation

- Welch average: XLALREAL8AverageSpectrumWelch()
- Median selection: XLALREAL8AverageSpectrumMedian()

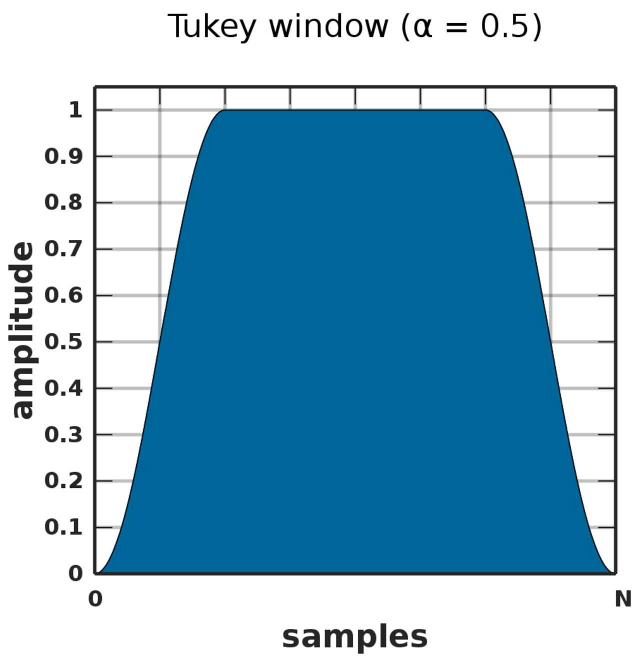


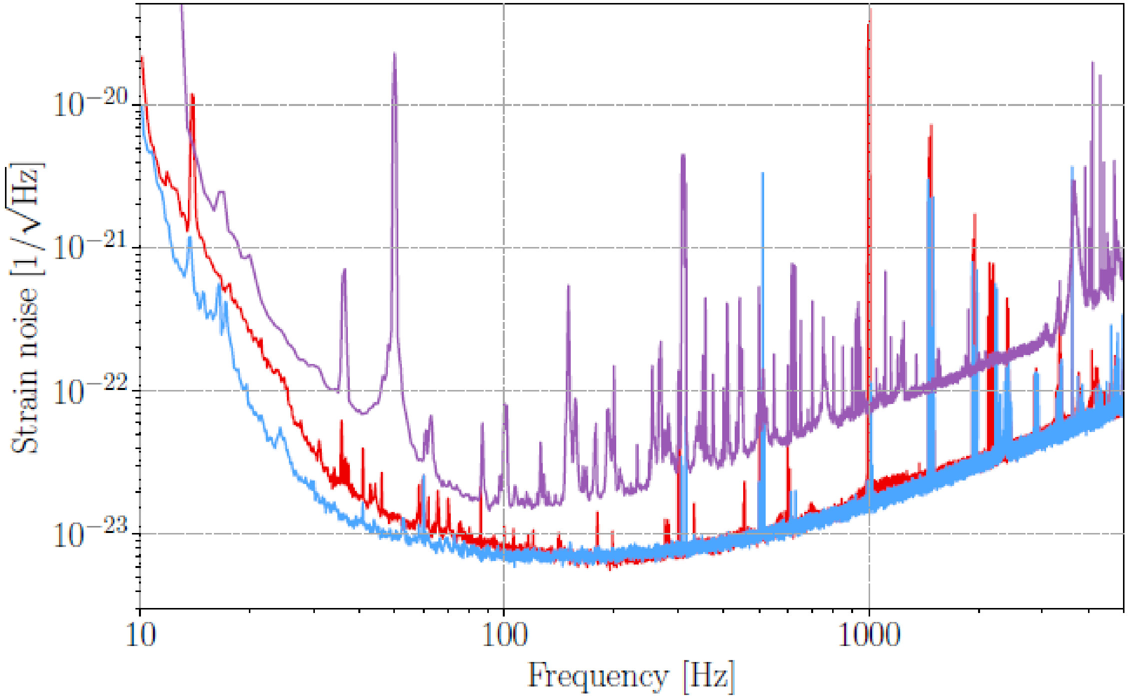
ASD of some LIGO noise



Data Analysis

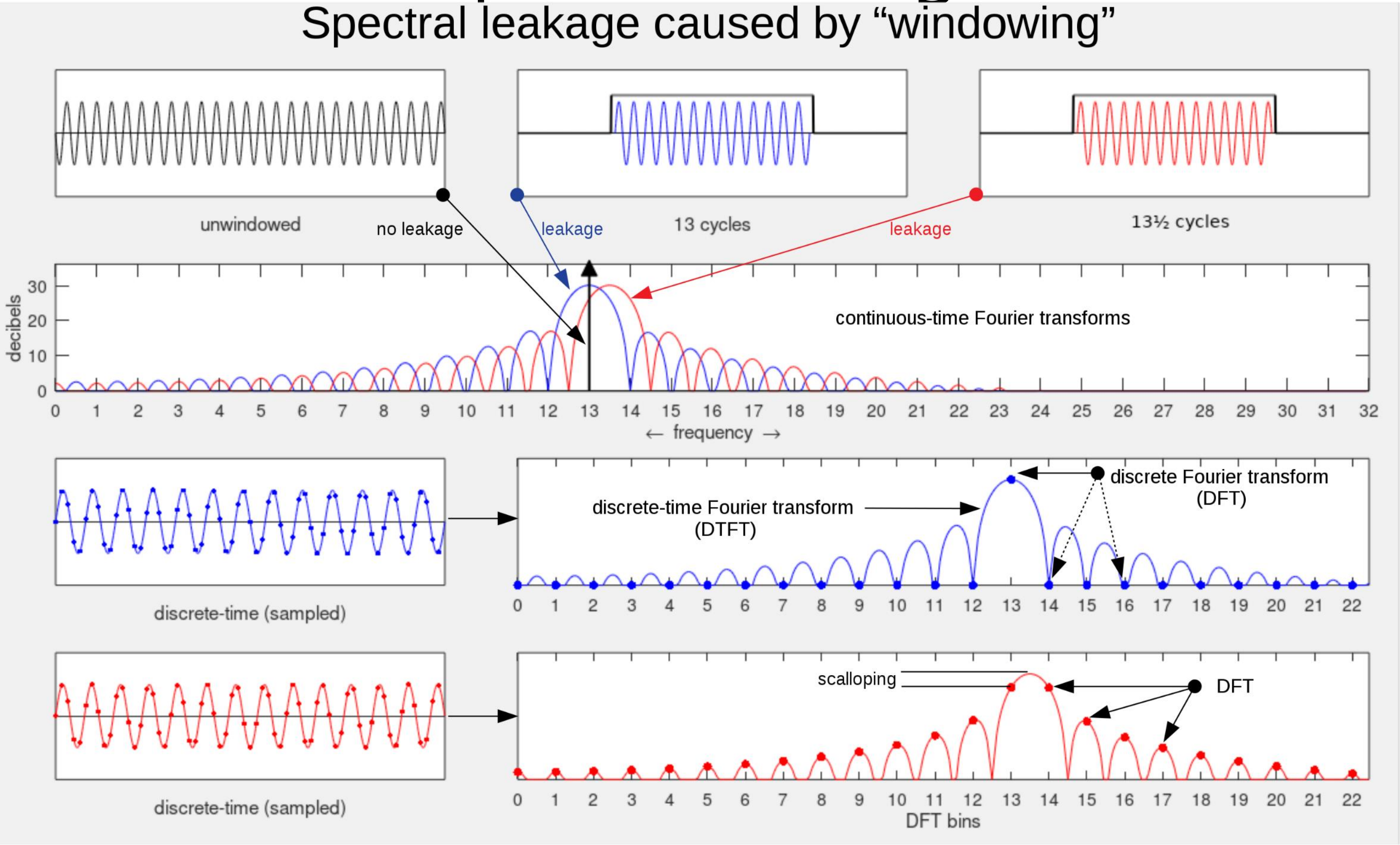
$$d(t) \xrightarrow{\text{FFT}} \tilde{d}(f) \xrightarrow{\text{Whiten}} \tilde{d}_w(f) = \frac{\tilde{d}(f)}{S_n^{1/2}(f)} \xrightarrow{\text{iFFT}} d_w(t)$$





Window and power leakage

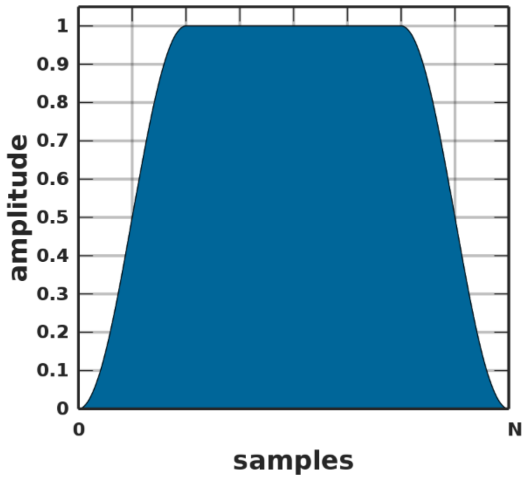
[Window function - Wikipedia](https://en.wikipedia.org/wiki/Window_function)



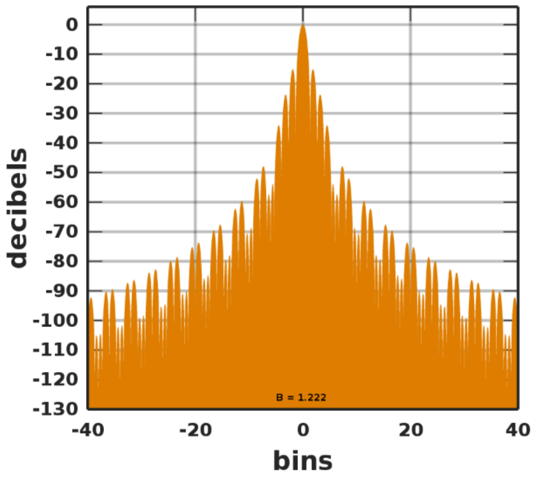
- GW150914

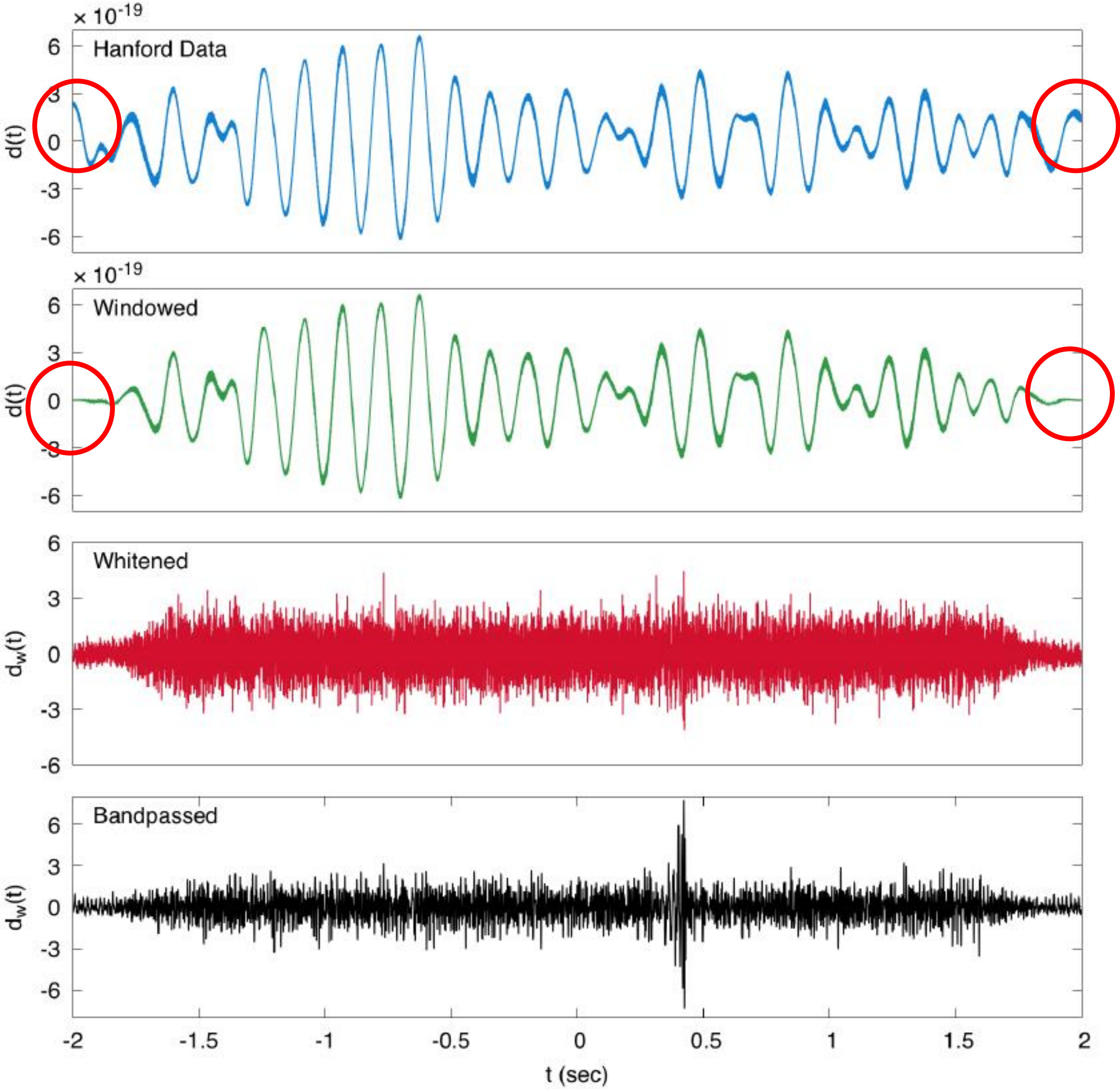
$$
\left.\begin{aligned}
w[n] &= \frac{1}{2}\left[1-\cos\left(\frac{2\pi n}{\alpha L}\right)\right], && 0 \le n < \frac{\alpha L}{2} \\
w[n] &= 1, && \frac{\alpha L}{2} \le n \le \frac{N}{2} \\
w[N-n] &= w[n], && 0 \le n \le \frac{N}{2}
\end{aligned}\right\}
$$

Tukey window (α = 0.5)

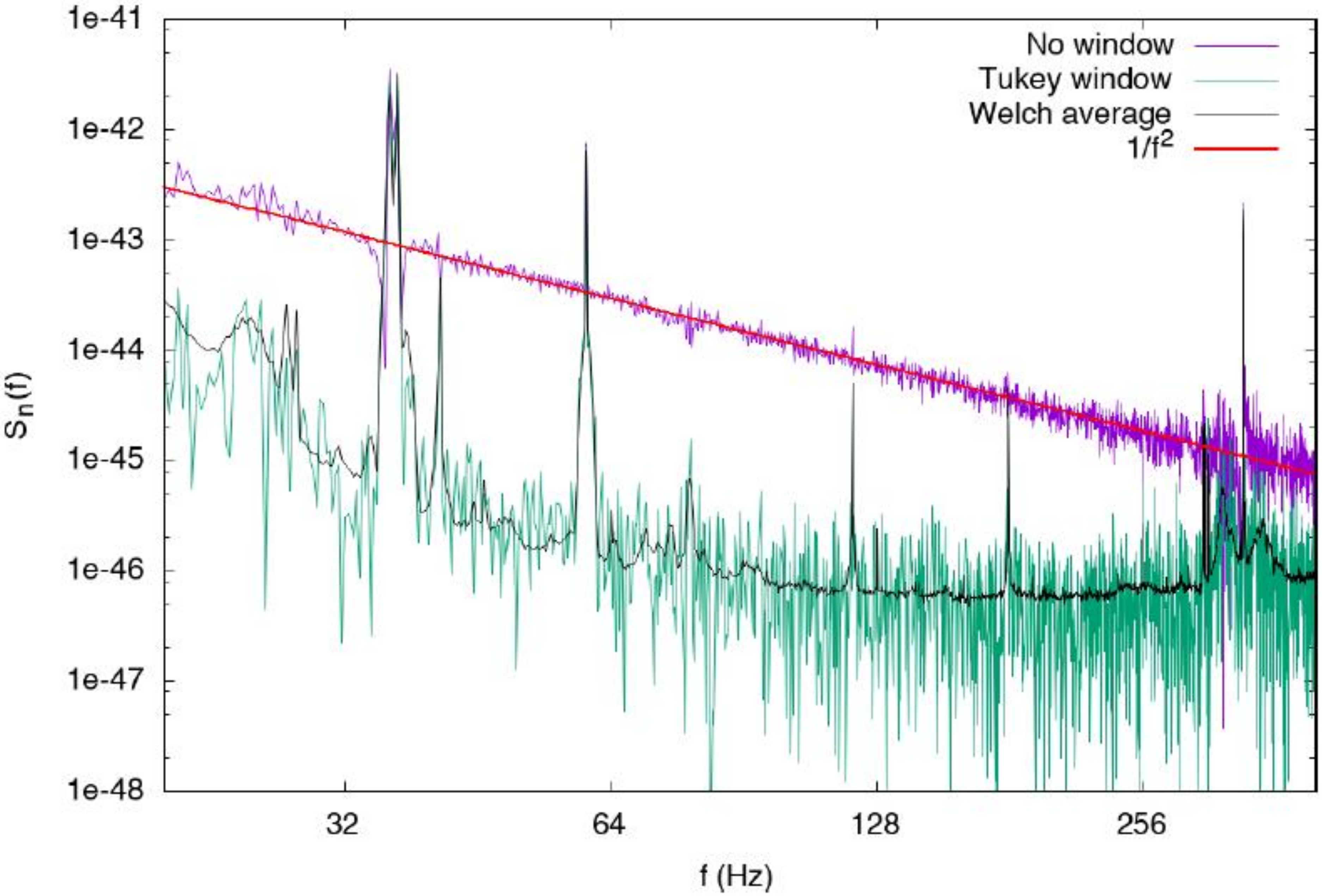


Fourier transform





Spectrun

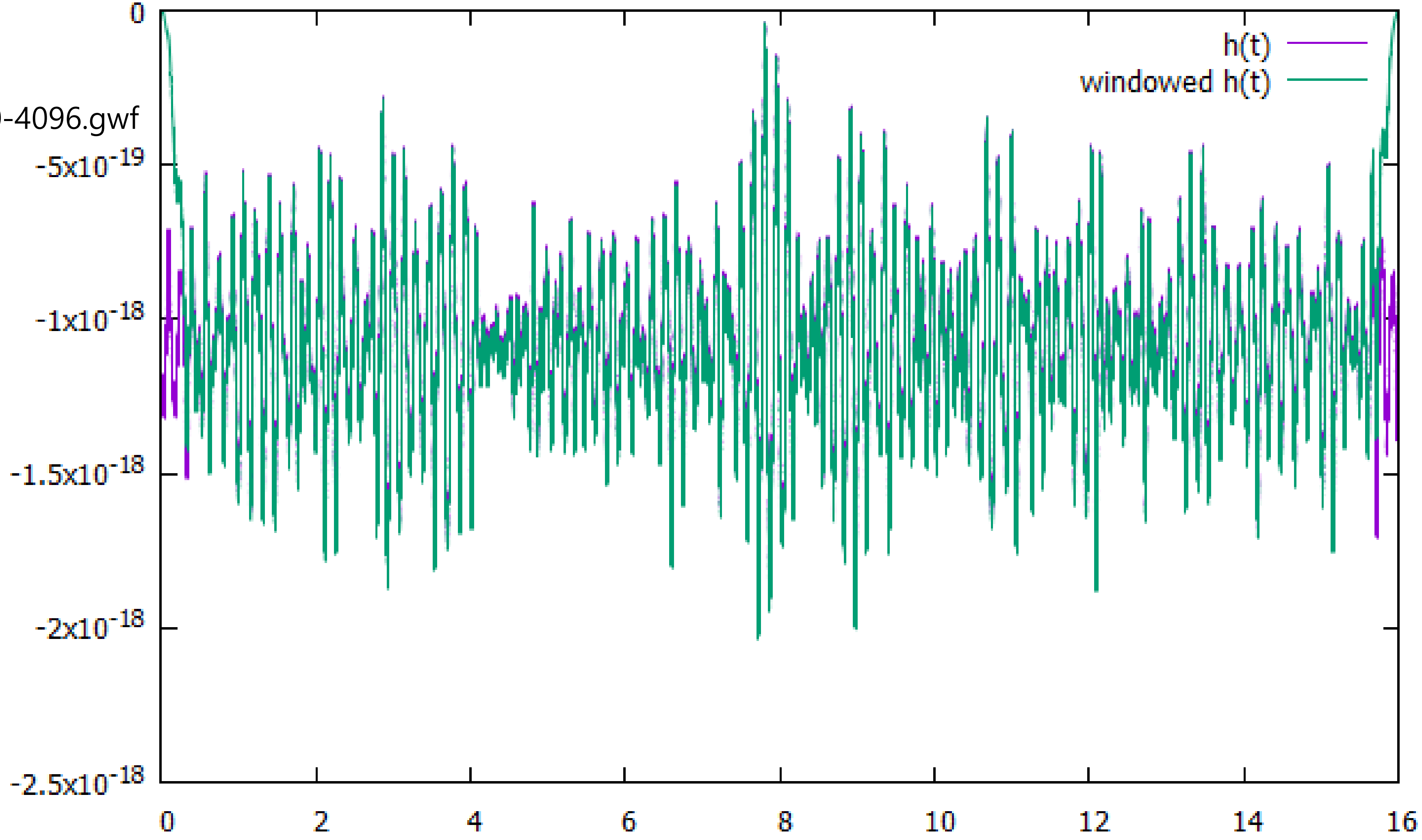


No window
Tukey window
Welch average
$1/f^2$
$S_n(f)$
1e-41
1e-42
1e-43
1e-44
1e-45
1e-46
1e-47
1e-48
32
64
128
256
f (Hz)

h(t) graph

KAGALI h(t) data

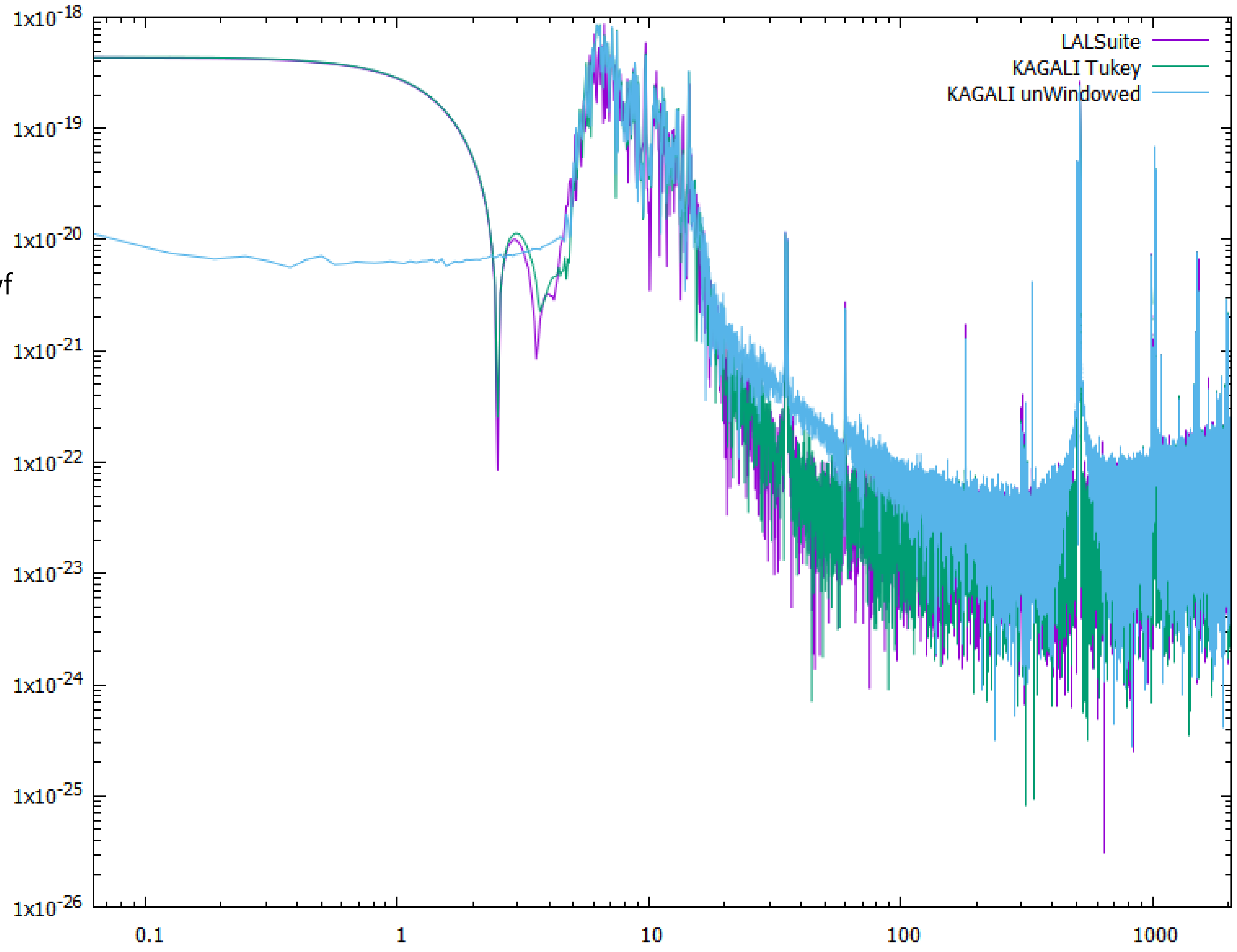
Window function : Tukey with 5% padding
Strain data : L-L1_LOSC_16_V1-1126256640-4096.gwf
Channel : L1:GWOSC-16KHZ_R1_STRAIN
Trigger time : 1126259462
Segment length : 16s



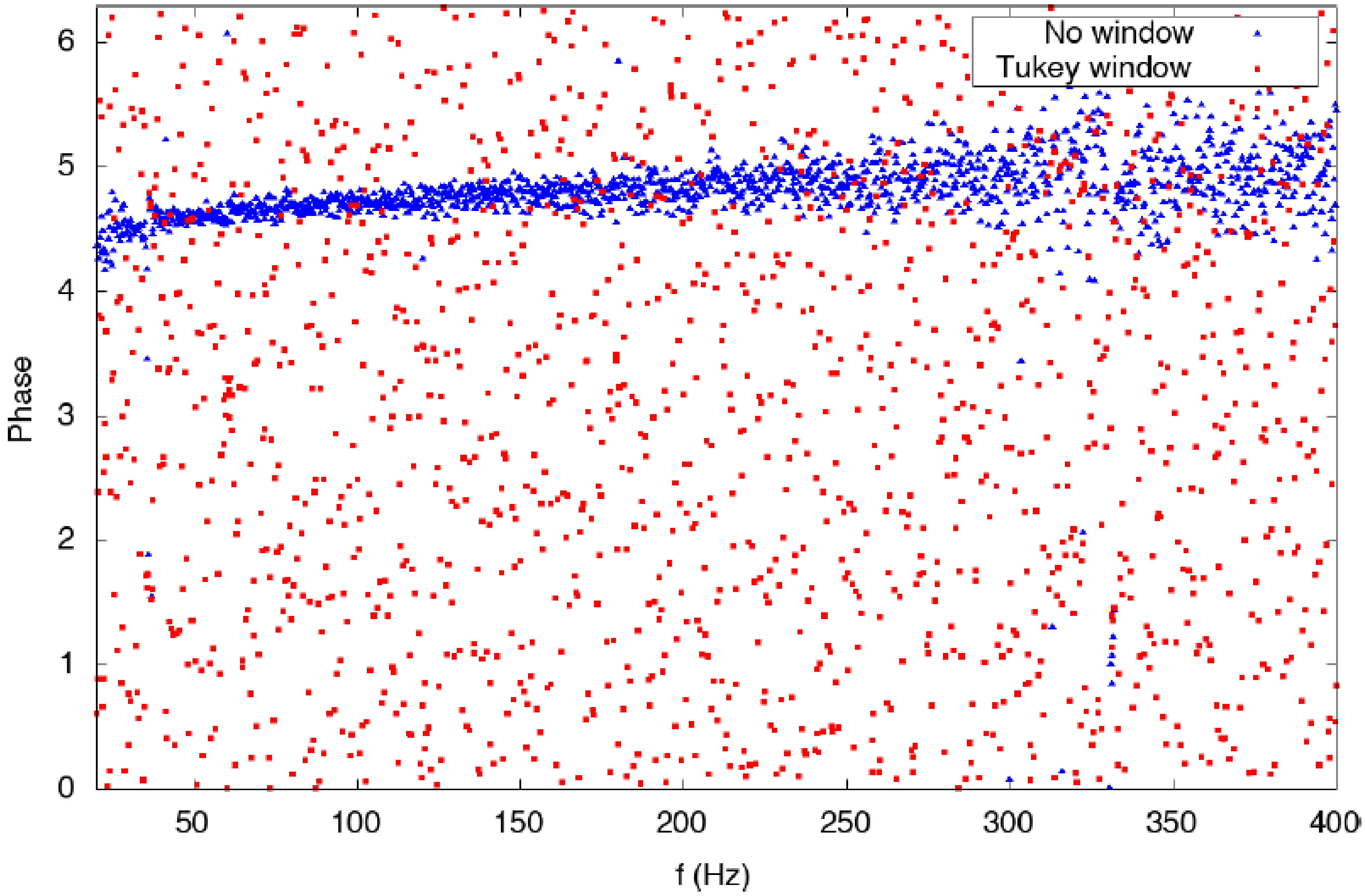
Spectrum

LALSuite and KAGALI h(f) comparison

Window function : Tukey with 0.4 sec padding
Strain data : L-L1_LOSC_16_V1-1126256640-4096.gwf
Channel : L1:GWOSC-16KHZ_R1_STRAIN
Trigger time : 1126259462
Segment length : 16s



Phase


No window
Tukey window
Phase
0
1
2
3
4
5
6
50
100
150
200
250
300
350
400
f (Hz)

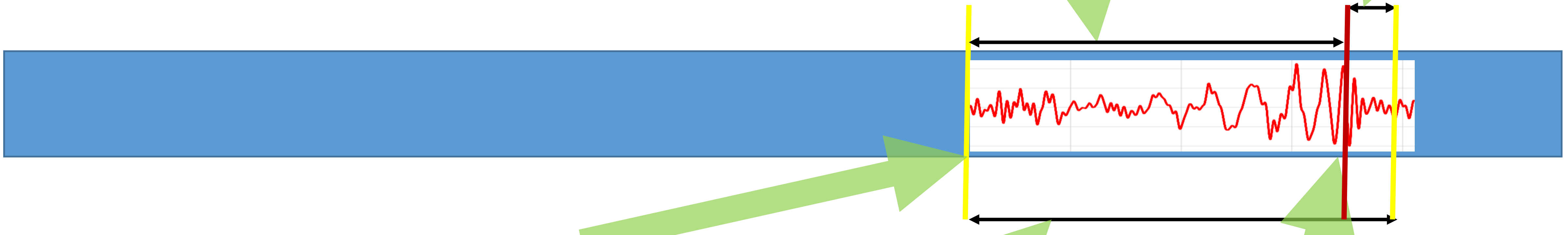
LALInferenceReadData.c

- LALInferenceReadData()
  - Read data file or generate fake data
  - Read PSD data file or generate from strain data
  - Resample as required sample rate
  - Windowing the time data
  - Generate frequency domain data using FFT
  - Generate whiten data in frequency and time

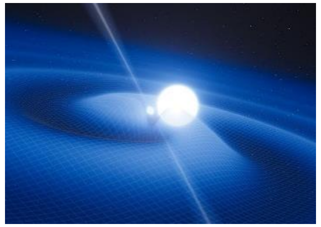
$$\boxed{d(t) \xrightarrow{\text{FFT}} \tilde{d}(f)} \xrightarrow{\text{Whiten}} \tilde{d}_w(f) = \frac{\tilde{d}(f)}{S_n^{1/2}(f)} \xrightarrow{\text{iFFT}} d_w(t)$$

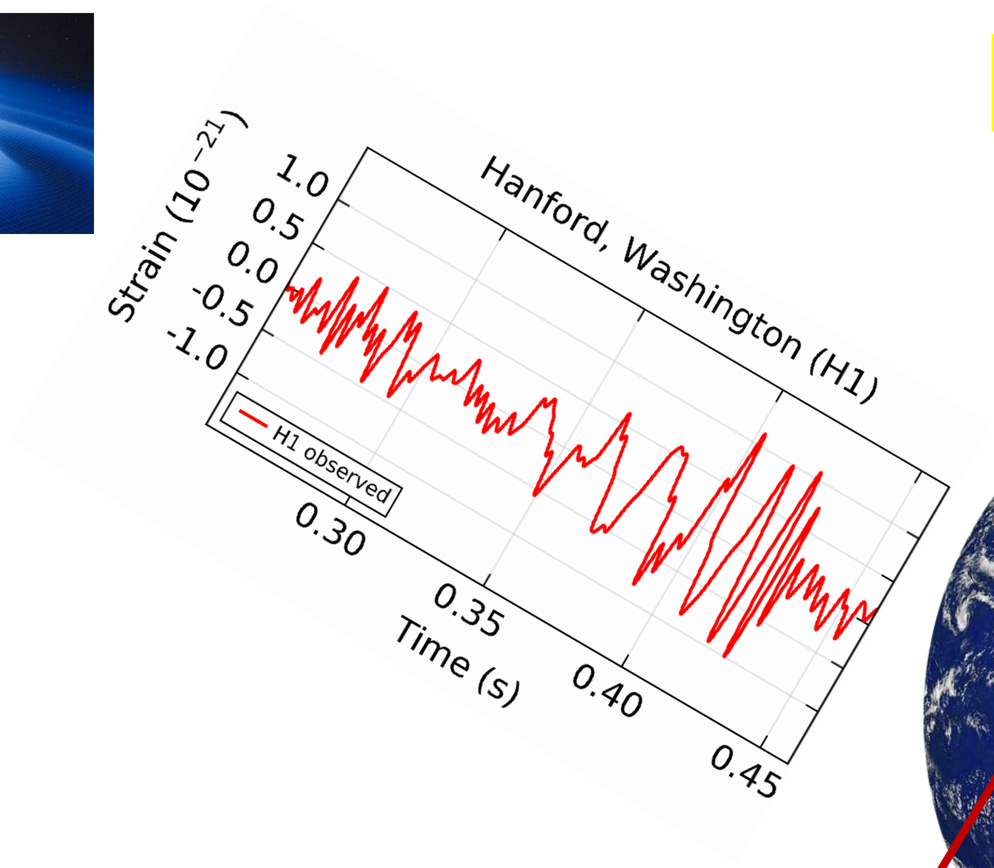
Time relations

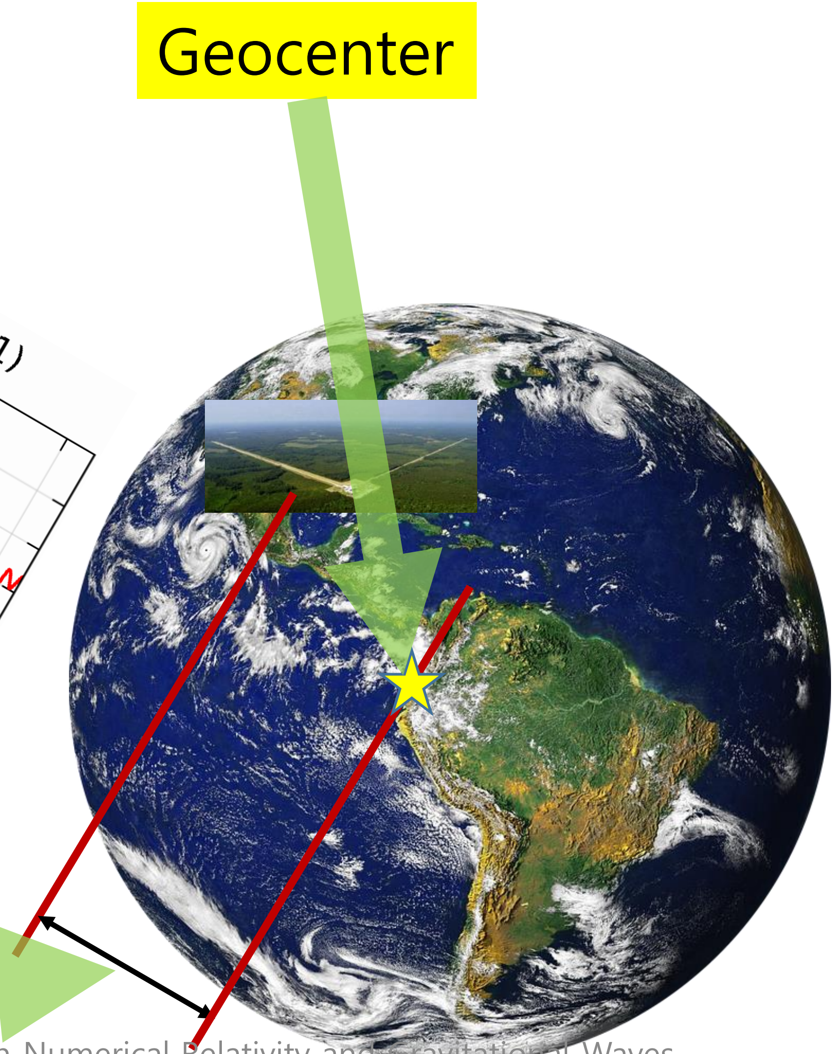
- --trigtime : This option specifies the trigger time at geocenter, this time is same as injection time in injection table
- ifo->epoch : this time is segment start time



Timedelay

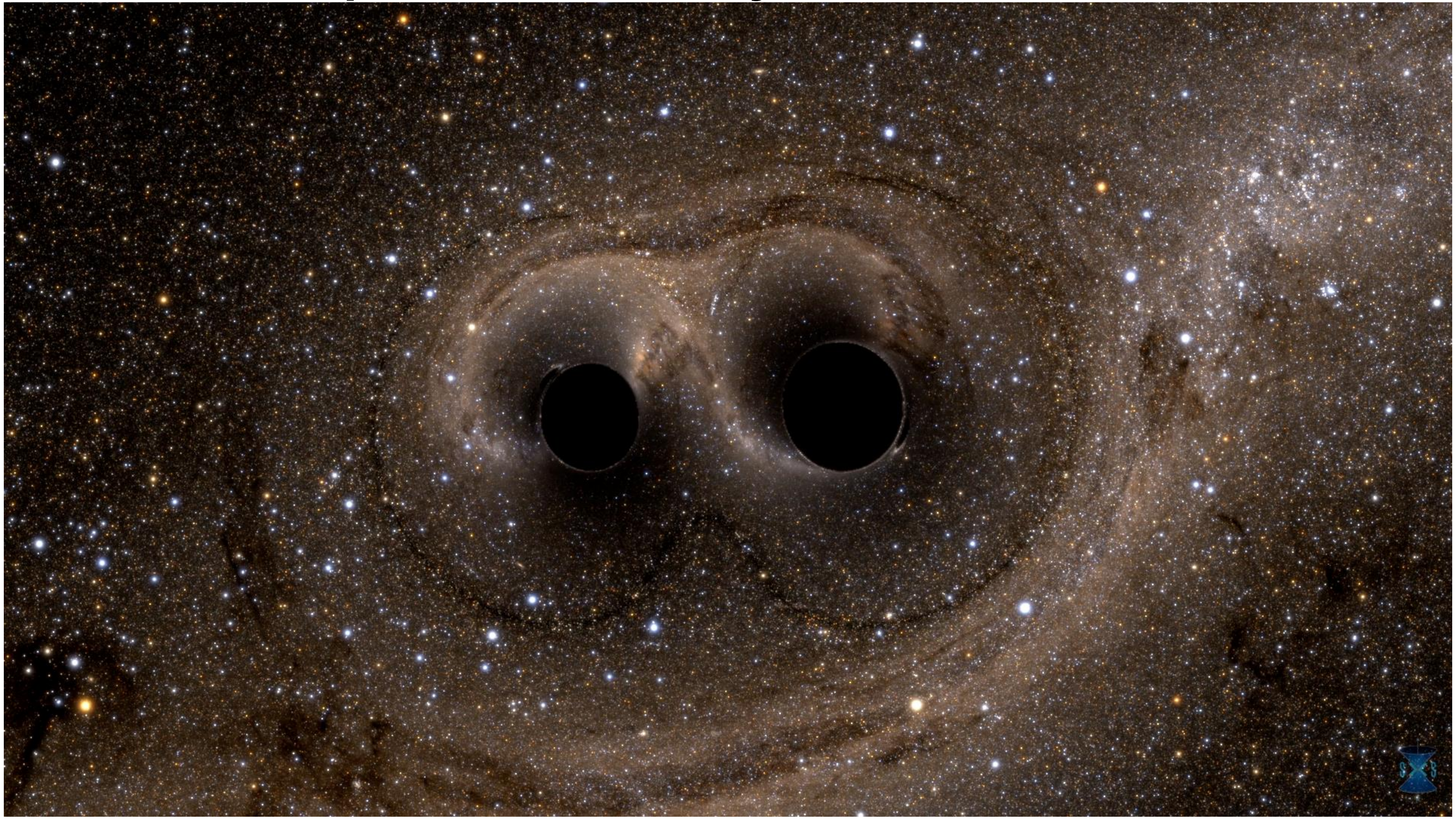



Hanford, Washington (H1)
Strain (10^{-21})
1.0
0.5
0.0
-0.5
-1.0
H1 observed
0.30
0.35
0.40
0.45
Time (s)

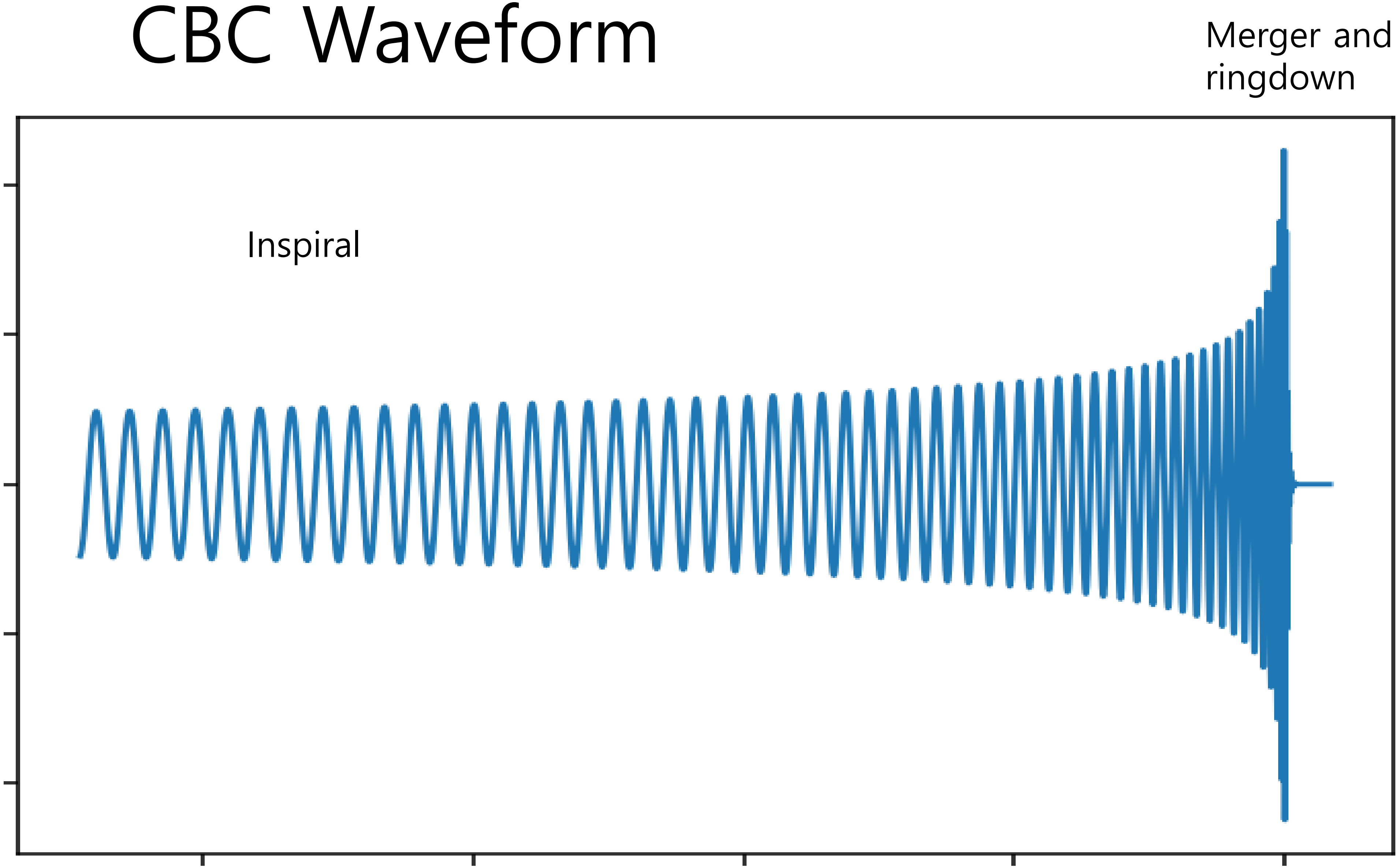

Geocenter
Timedelay

CBC Signal

CBC: Compact Binary Coalescence

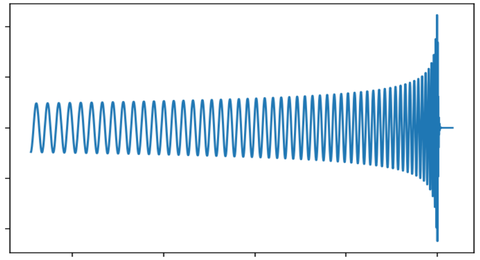


CBC Waveform



GW Frequency = 2X Orbital Frequency

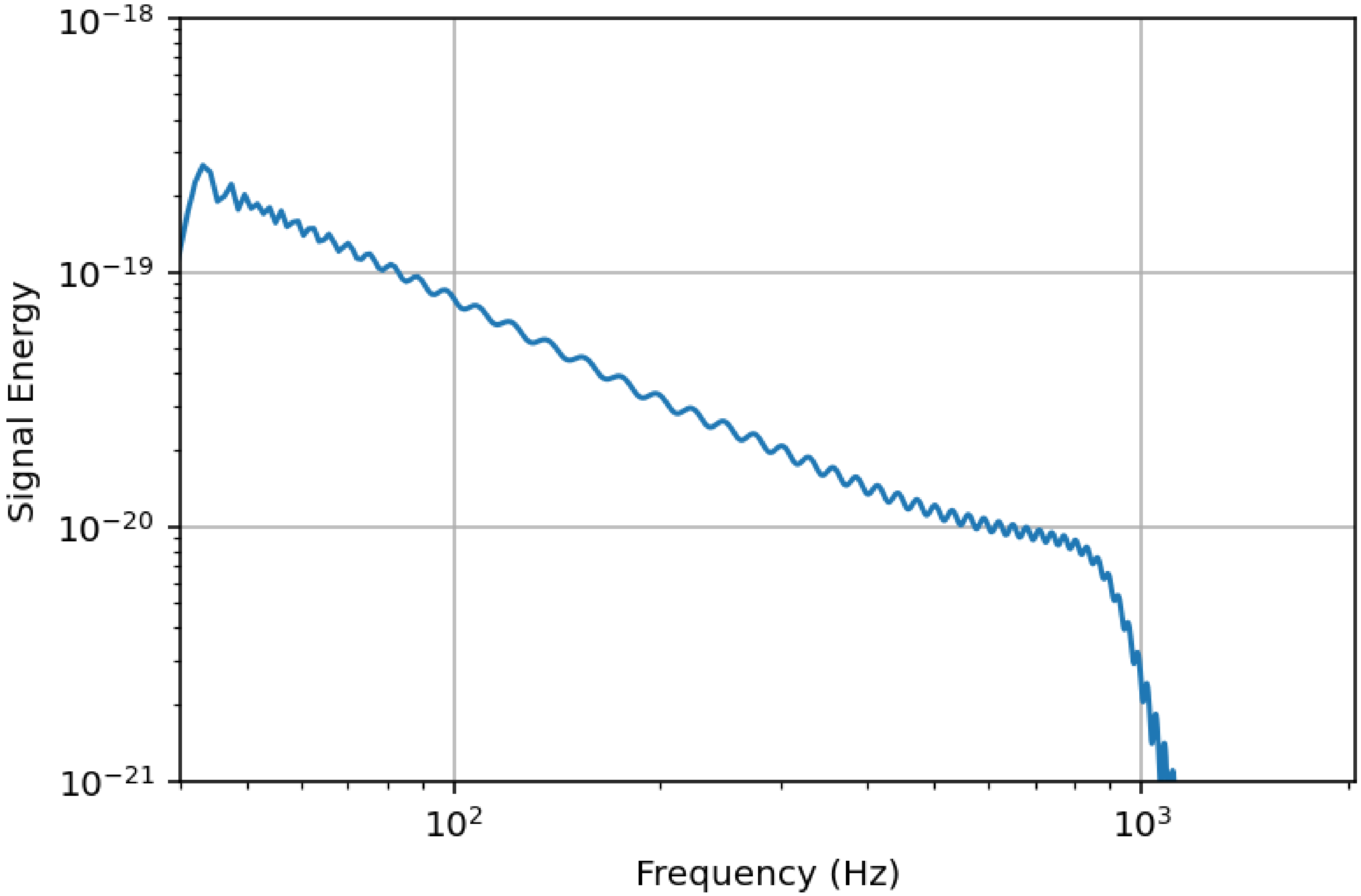
Frequency domain



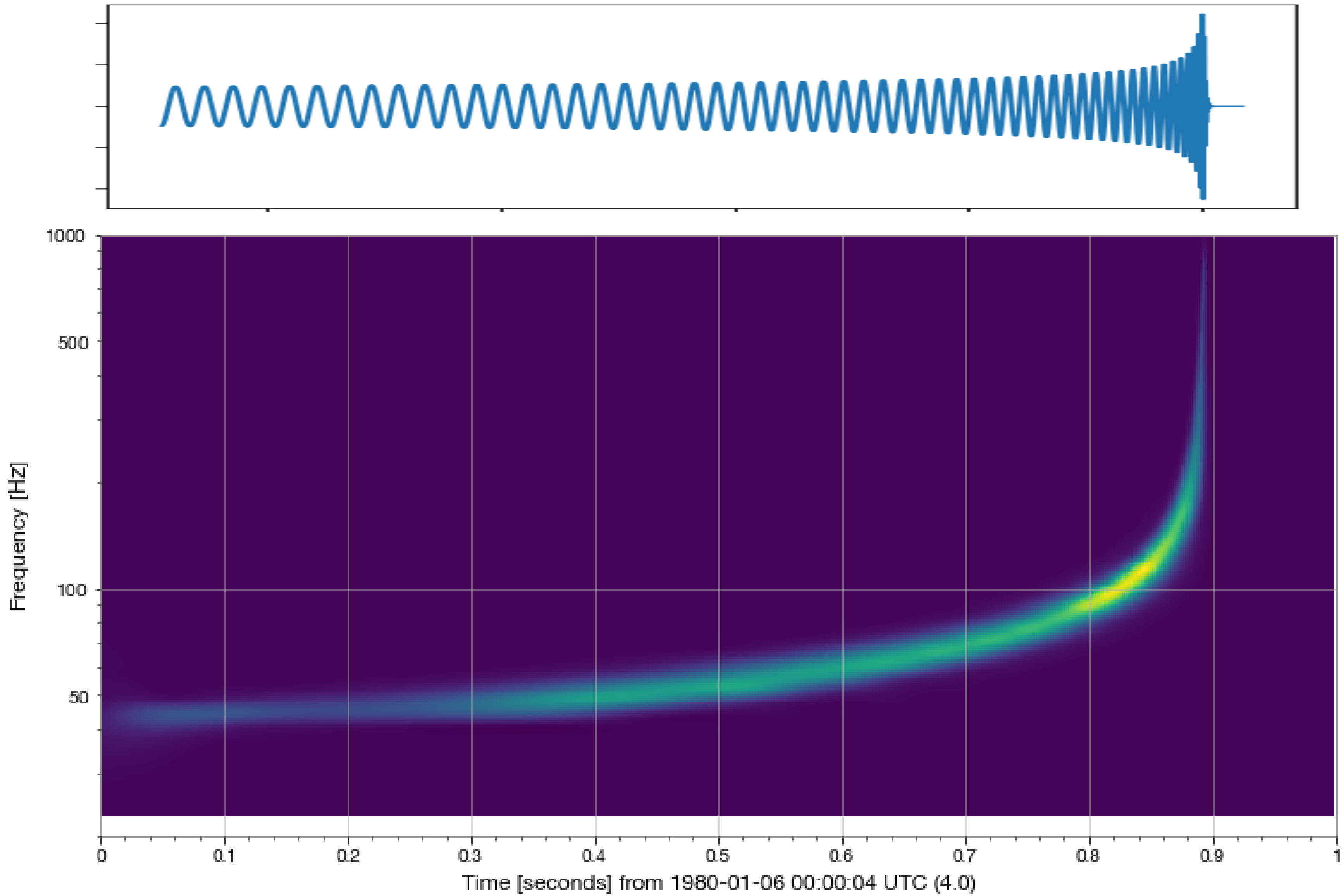
Lots of signal energy at low frequencies

Less signal energy at high frequencies

"Cut-off" around the merger frequency



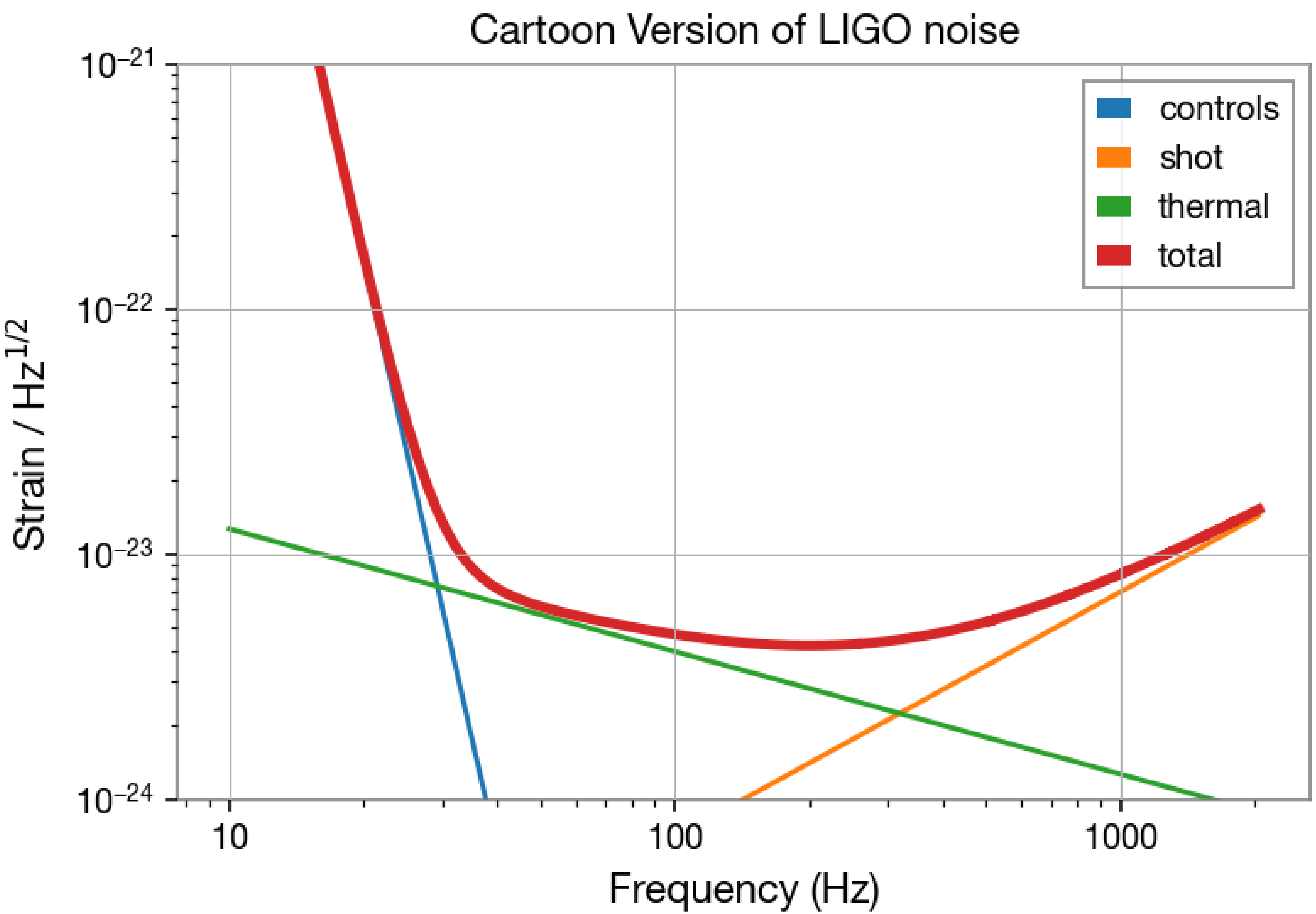
Spectrogram (Q-transform)



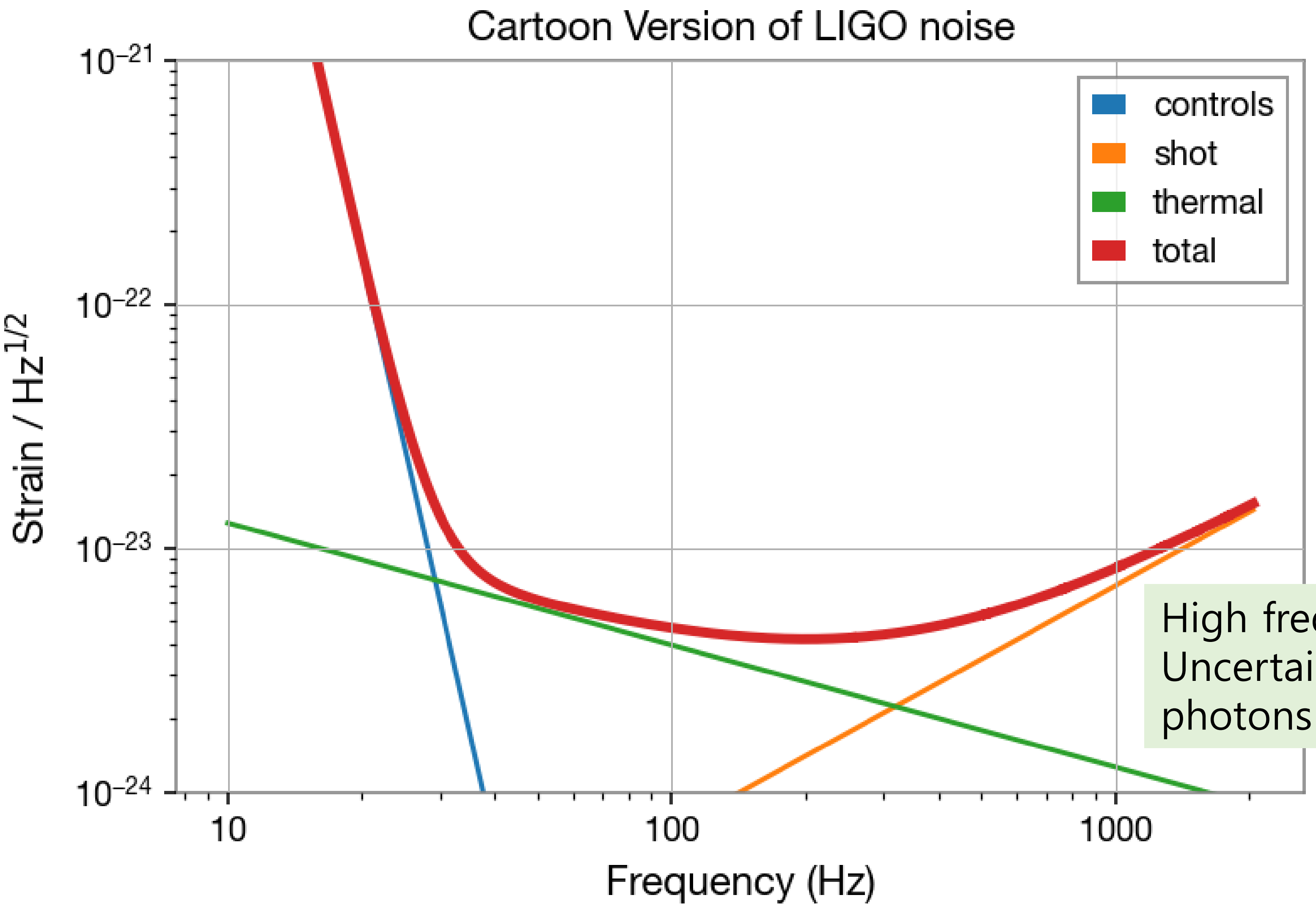
How long is the waveform?

- Black holes orbit more millions (or billions) of years.
- But, LIGO only “sees” events for a few seconds or less. Why?

LIGO Noise Cartoon

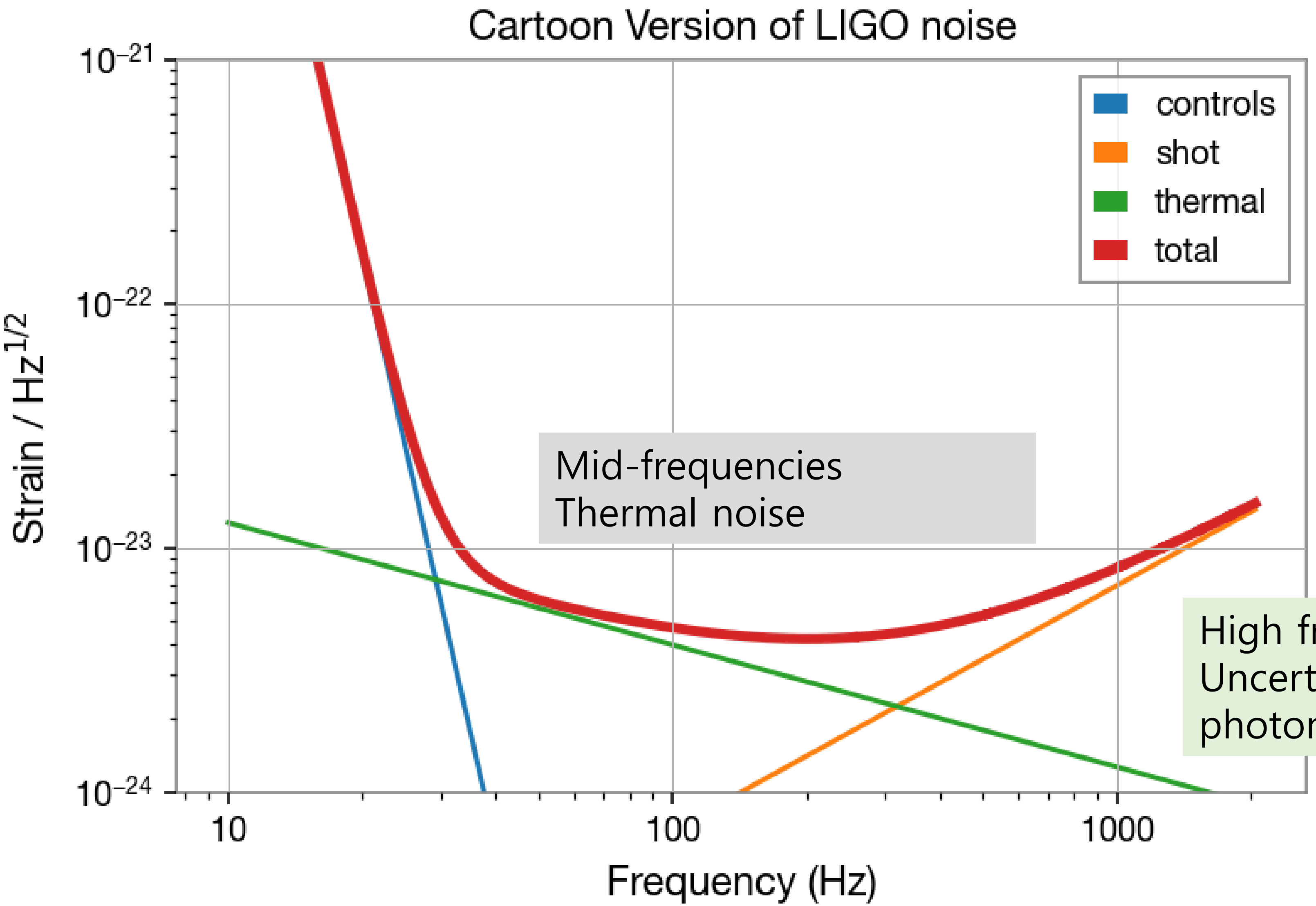


LIGO Noise Cartoon



High frequency noise
Uncertainty in number of photons

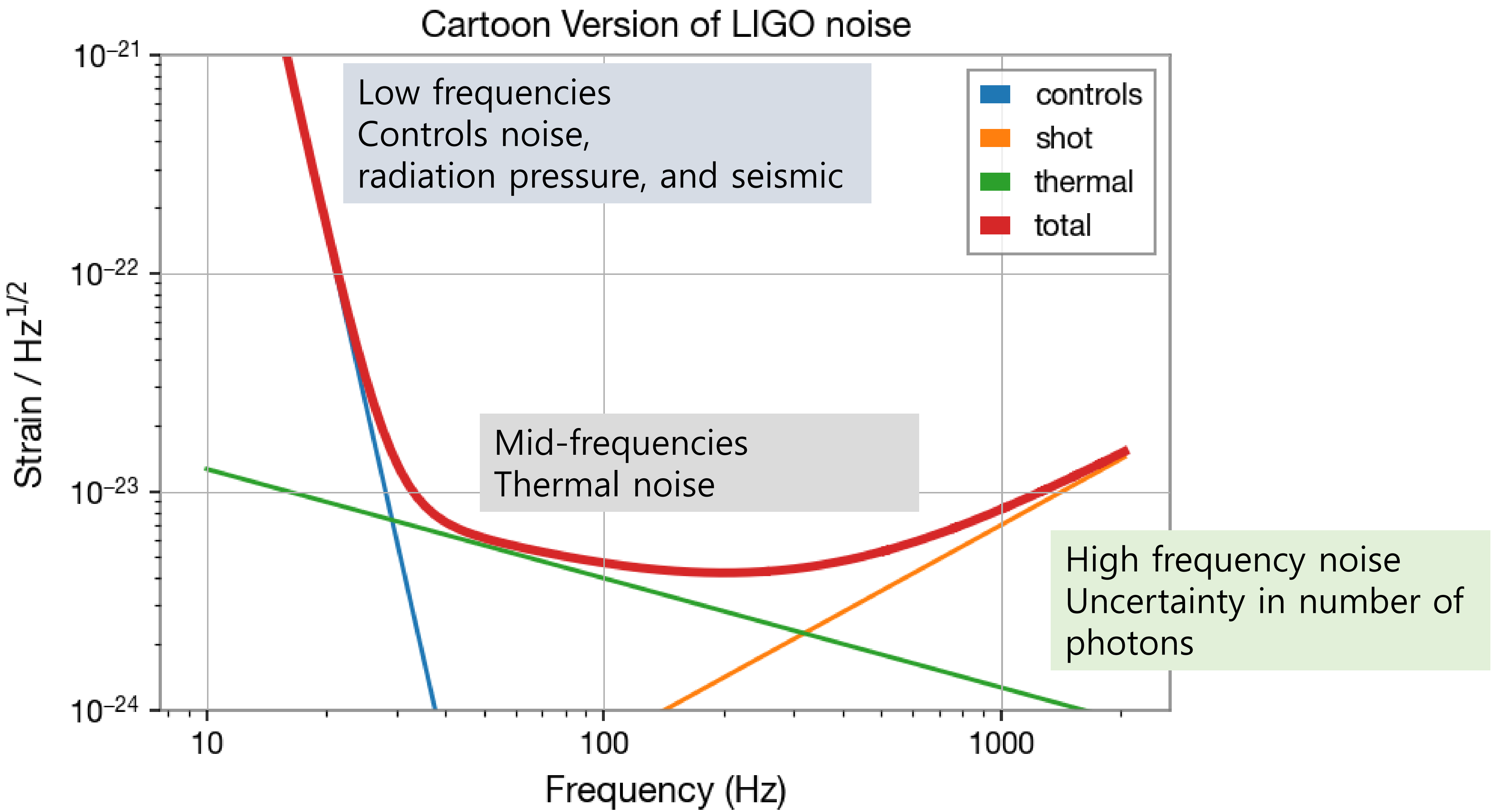
LIGO Noise Cartoon



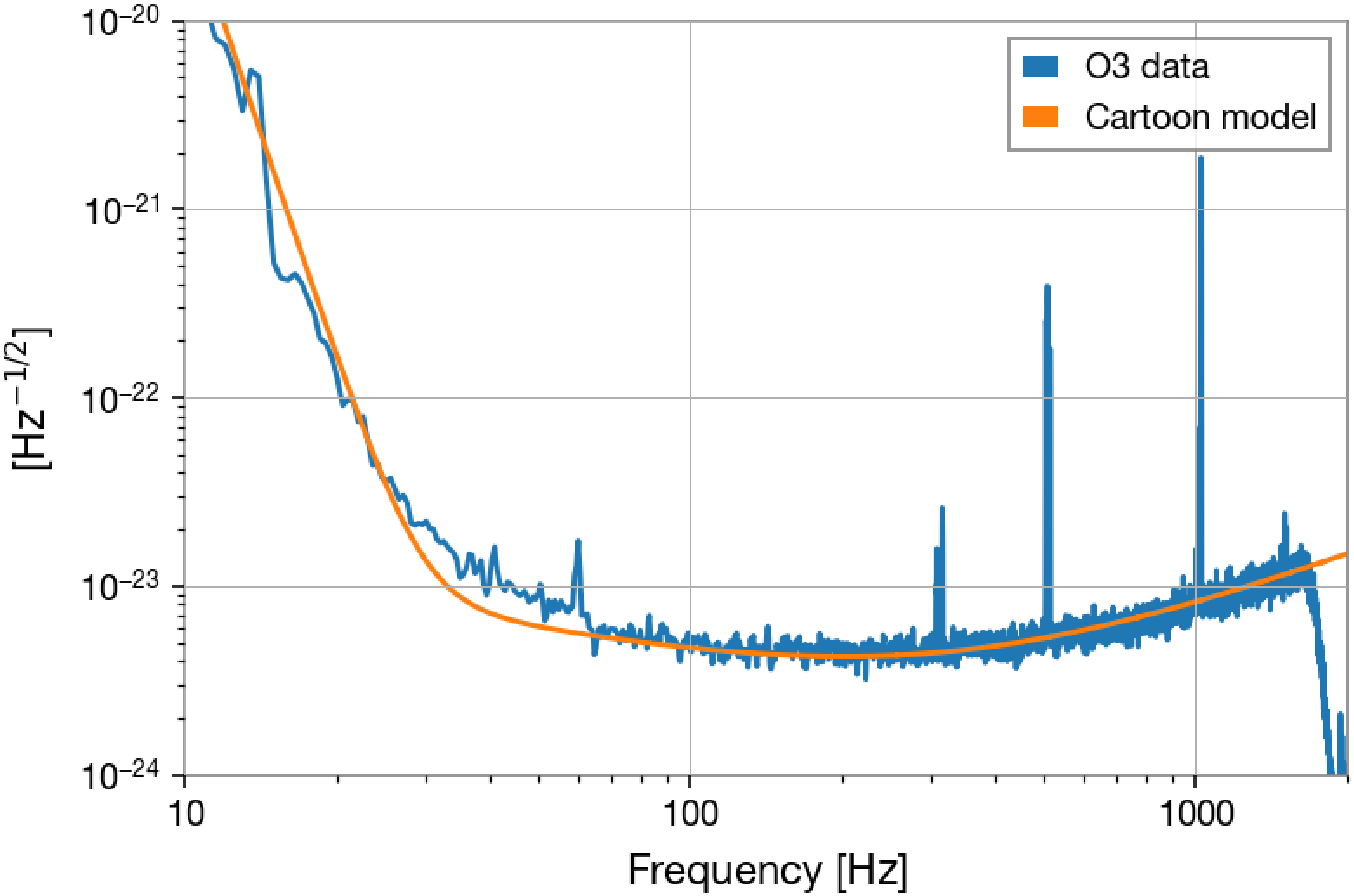
Mid-frequencies
Thermal noise

High frequency noise
Uncertainty in number of photons

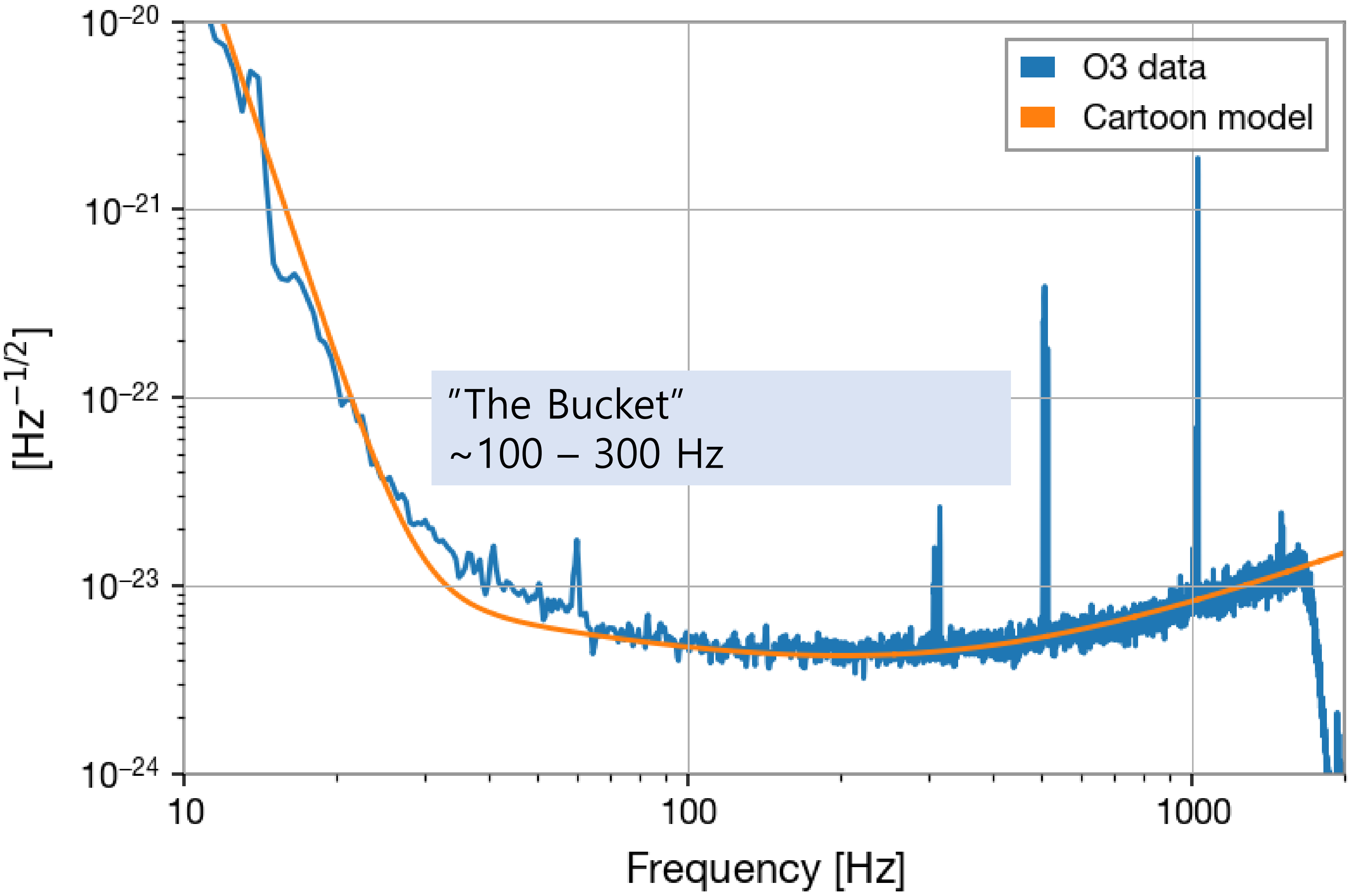
LIGO Noise Cartoon



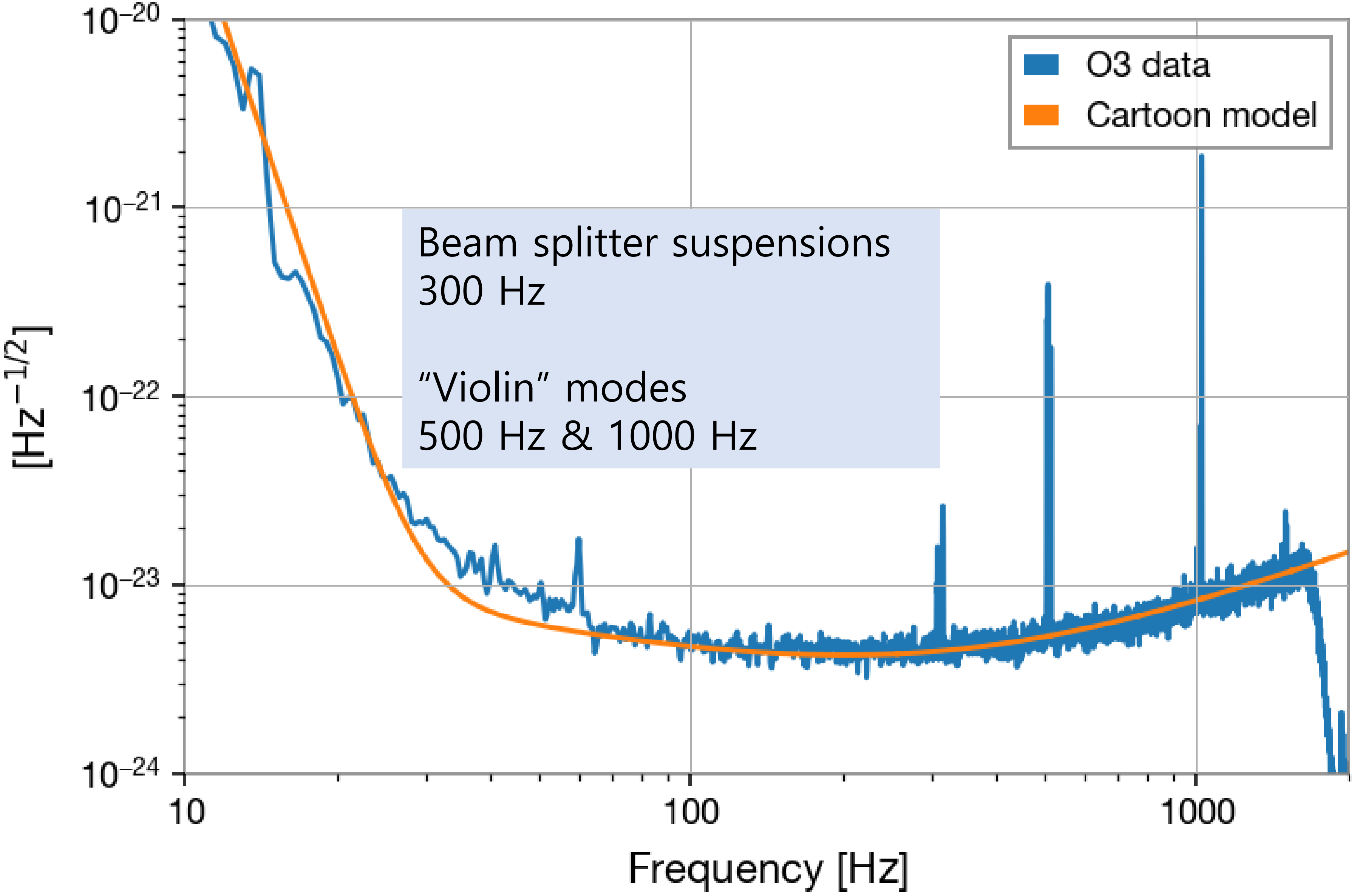
What frequency is LIGO most sensitive?



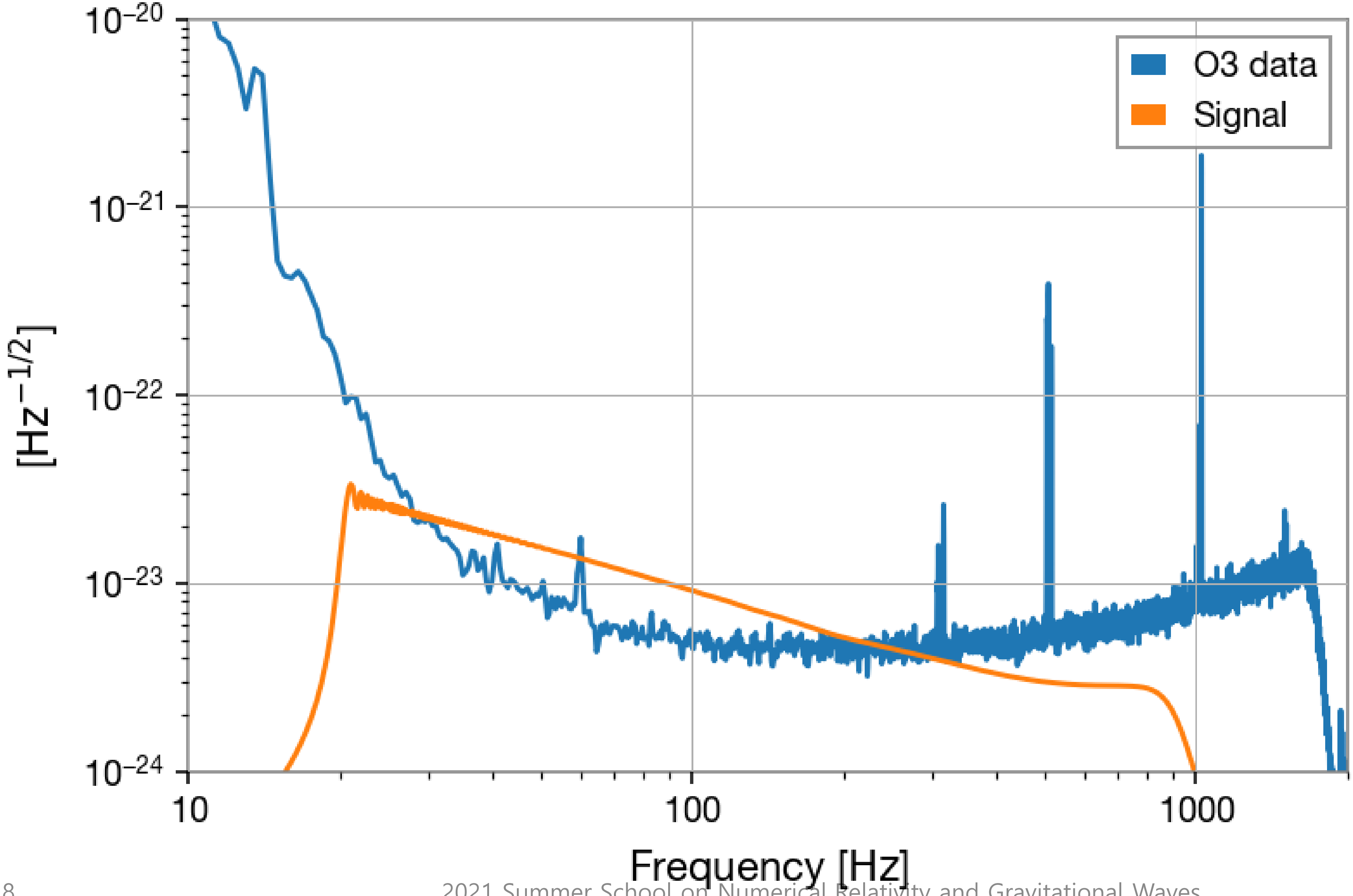
What frequency is LIGO most sensitive?

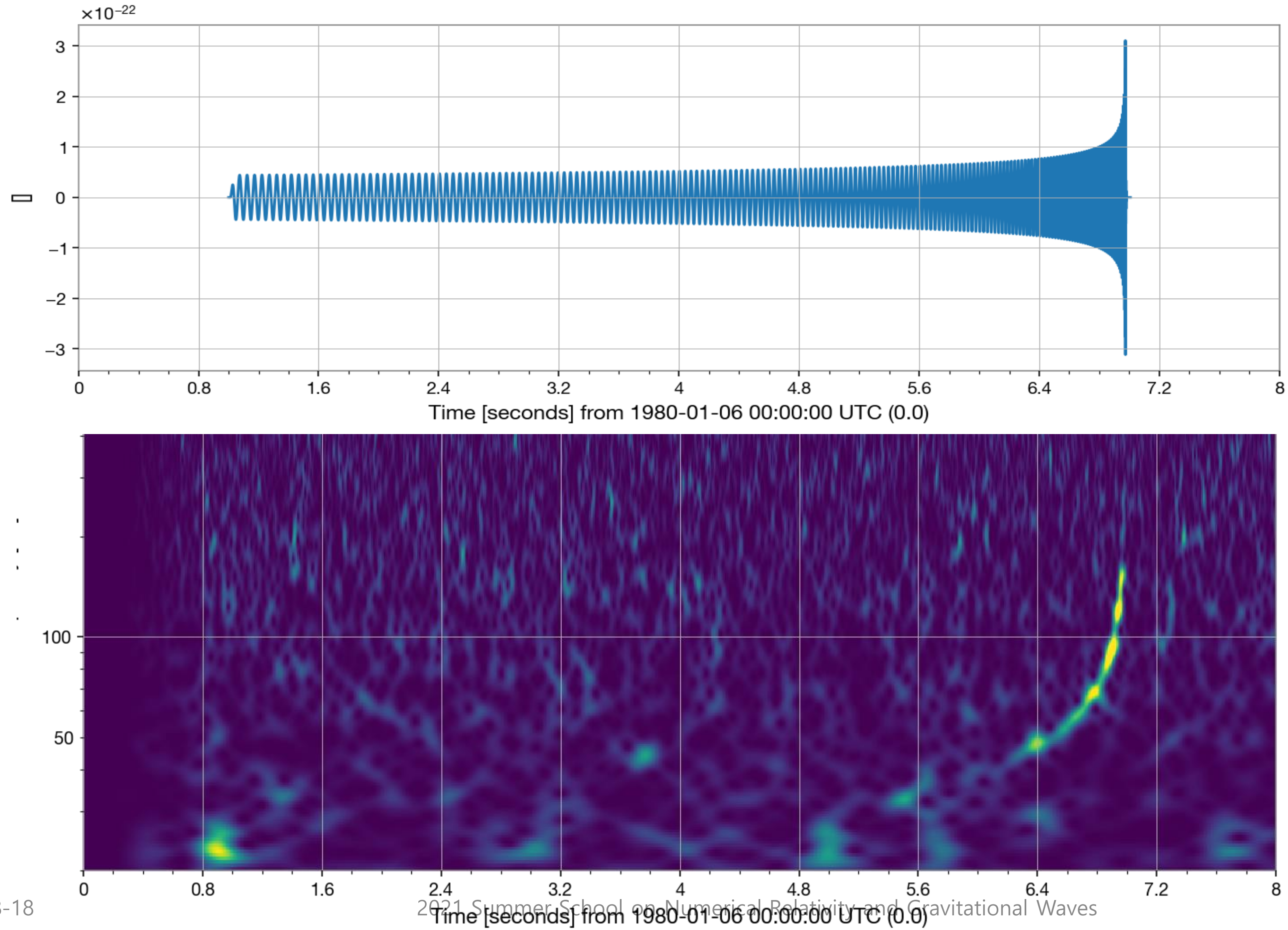


You can see resonant frequencies!



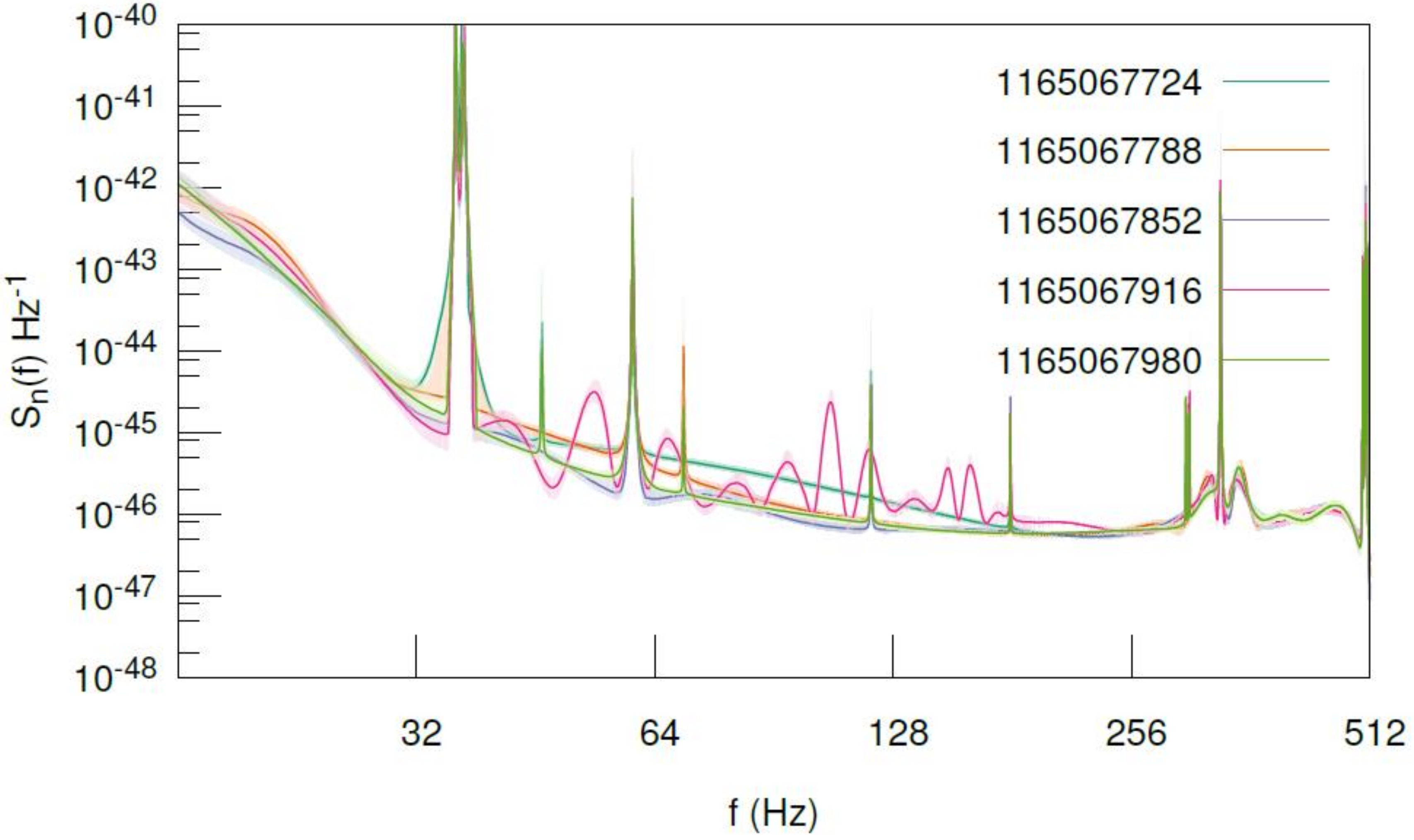
What happens at low frequencies?




×10⁻²²
3
2
1
0
−1
−2
−3
0
0.8
1.6
2.4
3.2
4
4.8
5.6
6.4
7.2
8
Time [seconds] from 1980-01-06 00:00:00 UTC (0.0)
100
50
0
0.8
1.6
2.4
3.2
4
4.8
5.6
6.4
7.2
8
Time [seconds] from 1980-01-06 00:00:00 UTC (0.0)

Time-frequency analysis and stationary

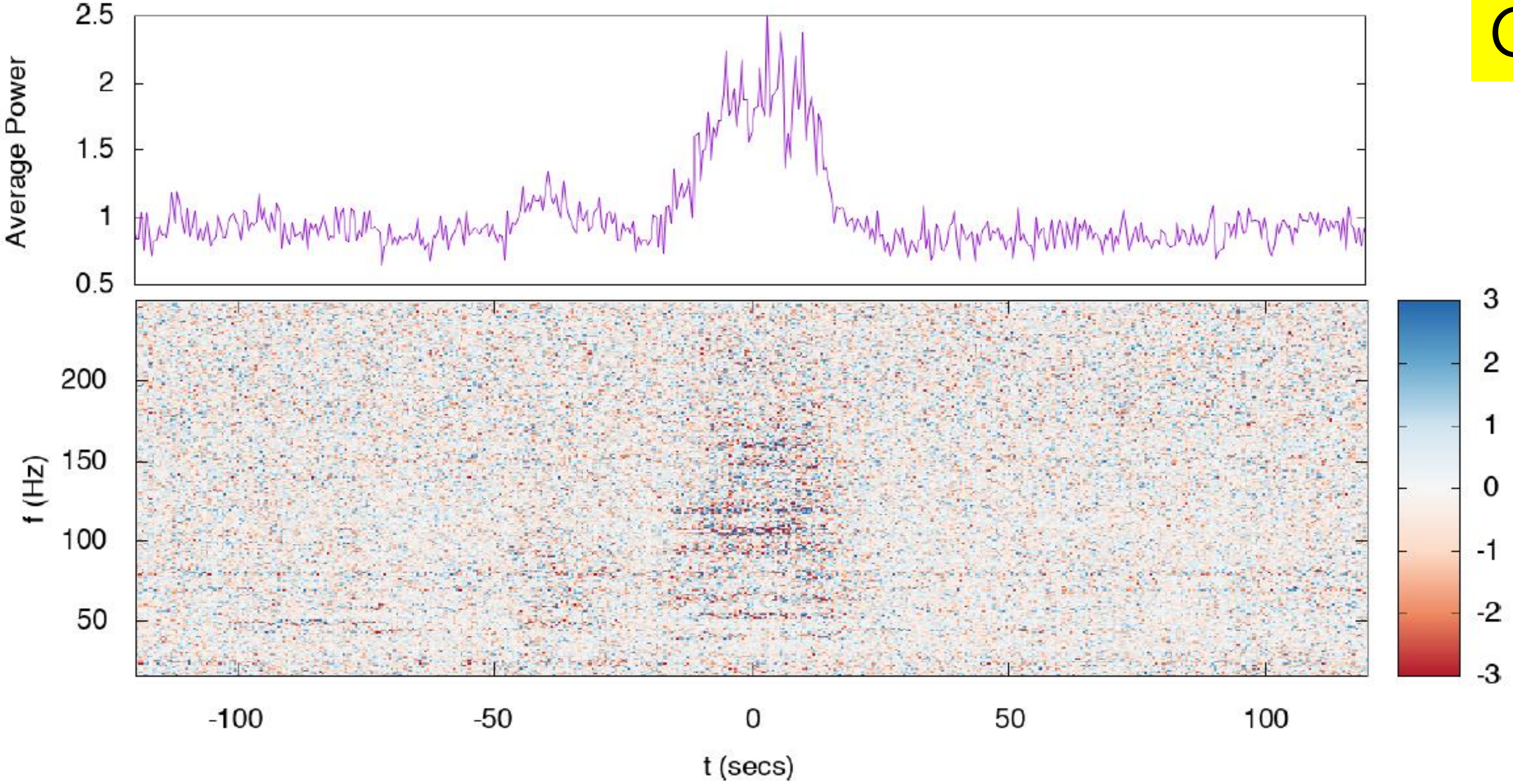
- Long-term
- Short-term: glitches
- AC lines



Stationarity test

- Continuous Wavelet transform(scalogram)

GPS 1165067917



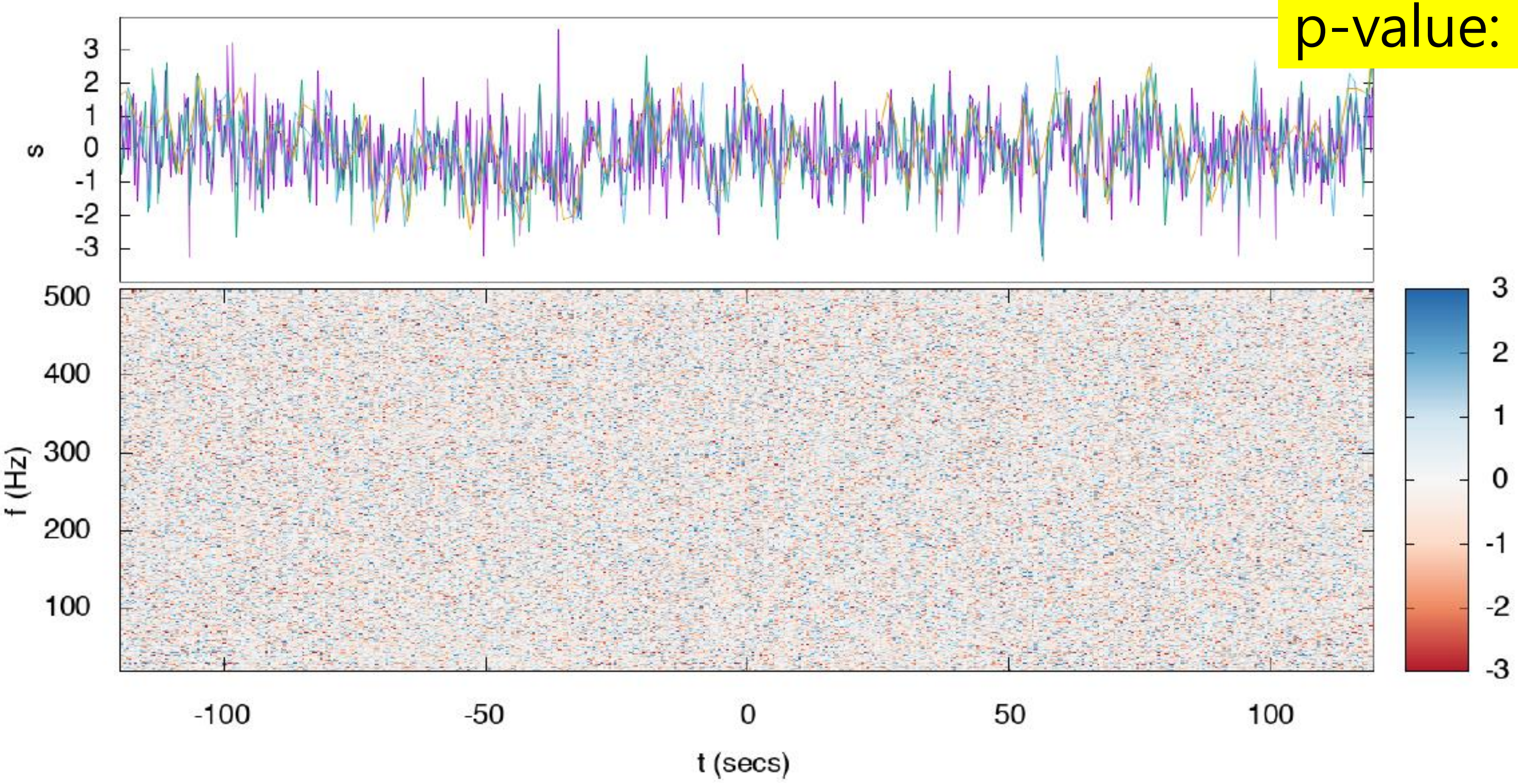
Wilson-Hilferty transfrom

- Transform Chi-square of many degree of freedom to standard one

$$W(Y) = \frac{\left(\frac{Y}{n}\right)^{1/3} - \left[1 - \left(\frac{1}{9}\right)\left(\frac{2}{n}\right)\right]}{\sqrt{\left(\frac{1}{9}\right)\left(\frac{2}{n}\right)}}$$

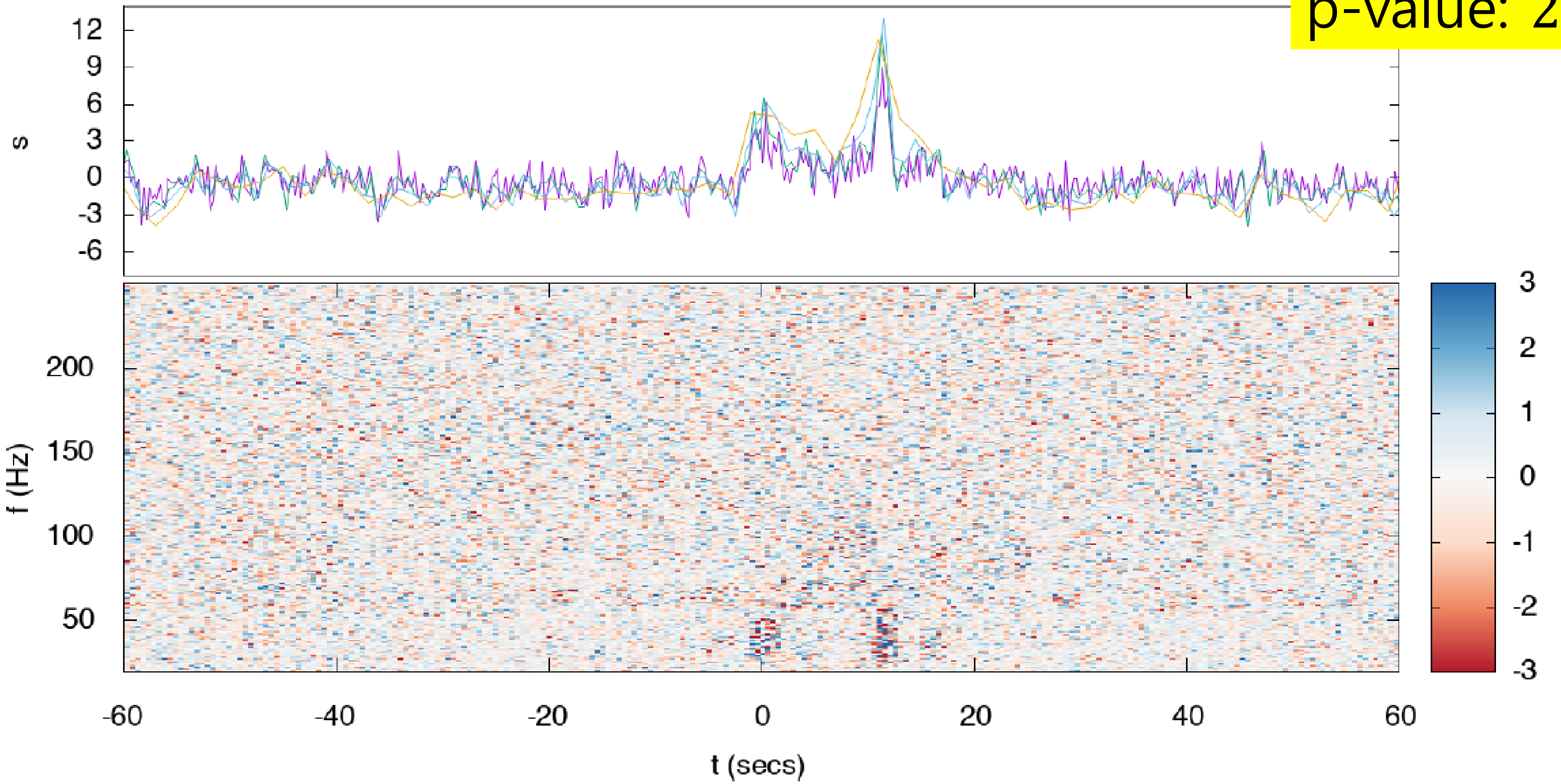
Wilson-Hilferty transform

GPS 1186741733
p-value: 0.74



Wilson-Hilferty transform

GPS 1166358283
p-value: 2.6×10^{-6}

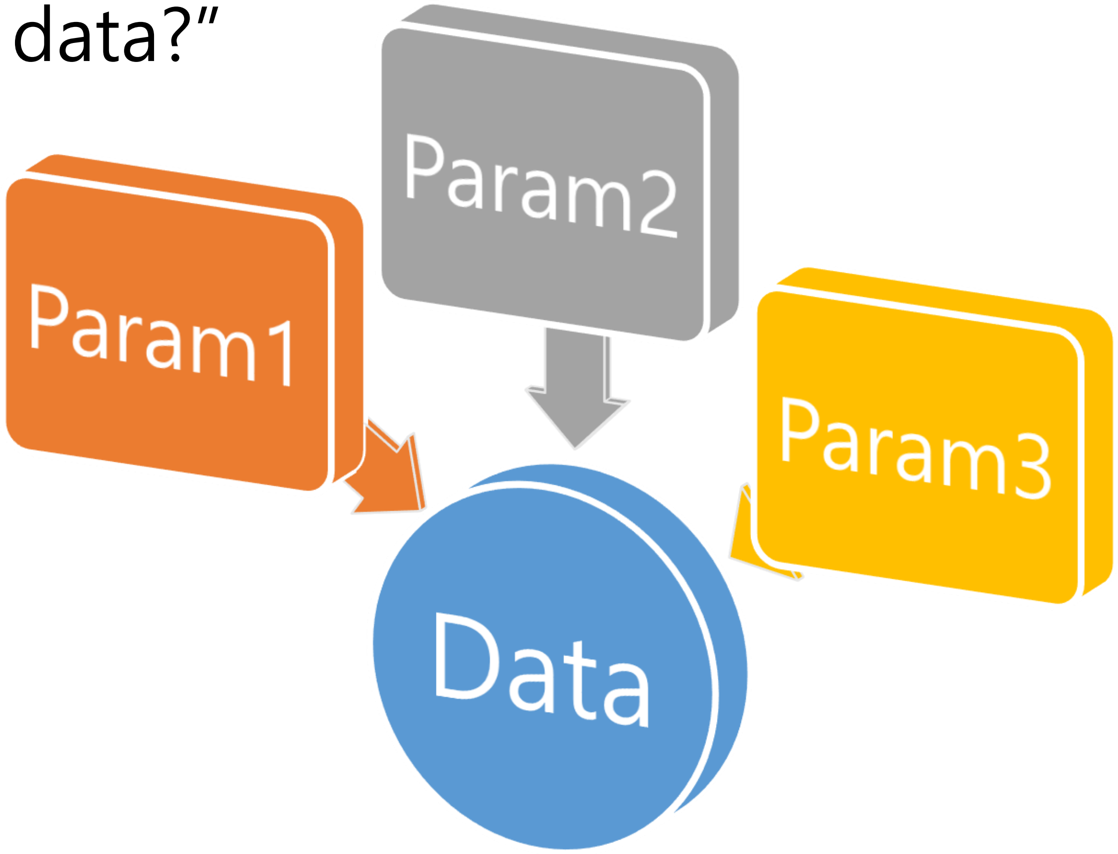


Parameter Estimation

Bayesian Inference

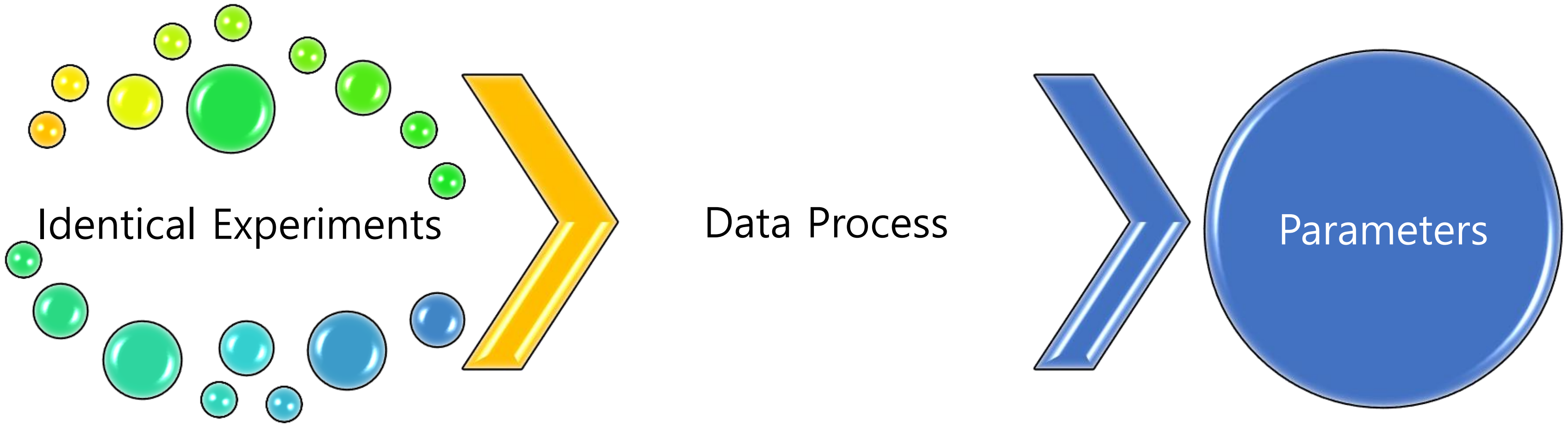
Bayesian Method

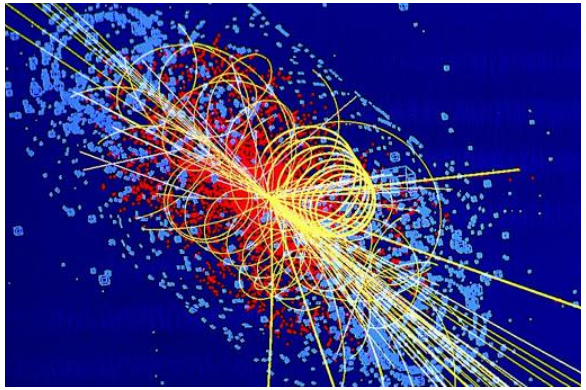
- Question: “What is the distribution of parameters for the given observed data?”

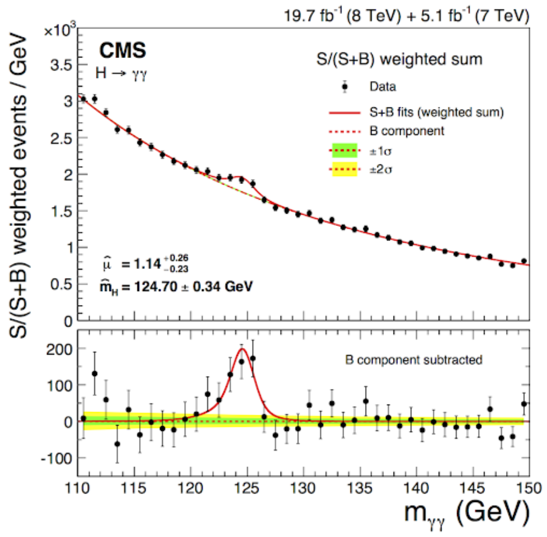


Experiment on Laboratory

- Question: “What is the distribution of parameters from data taken with many identical experiments?”



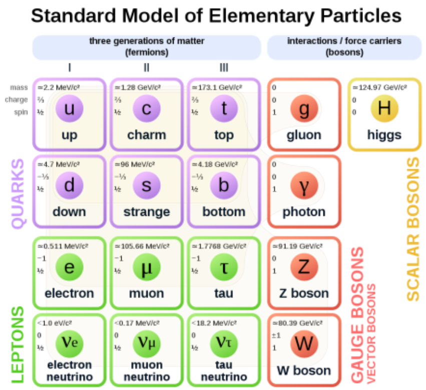


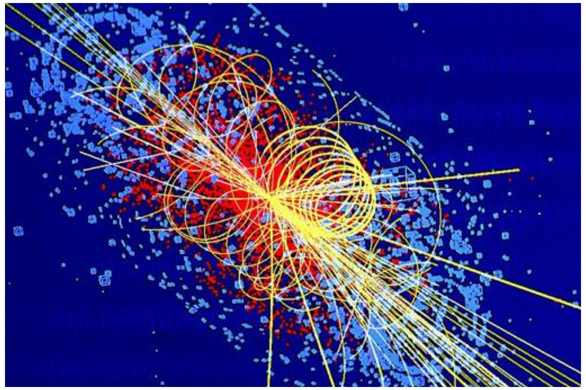


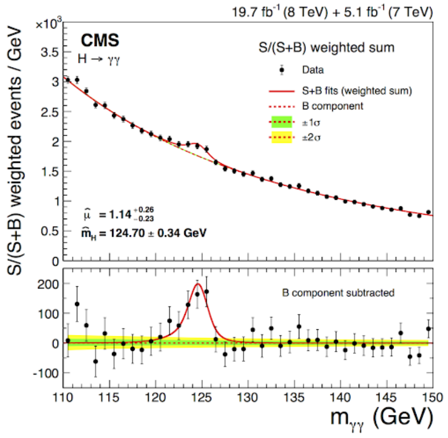
Classical vs. Bayesian Statistical Inference

Classical Inference

Model → Experiments → Confidence

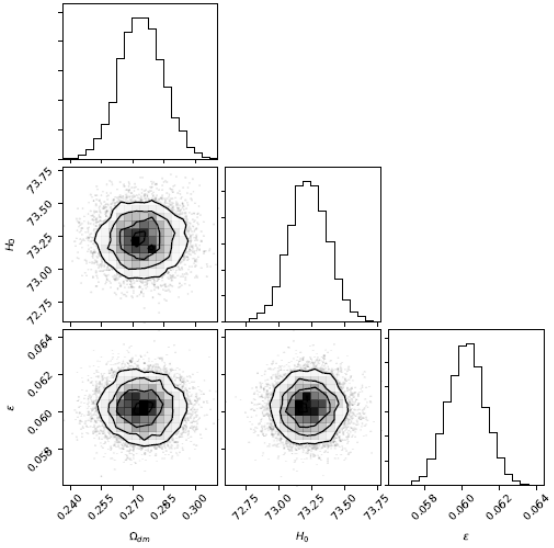


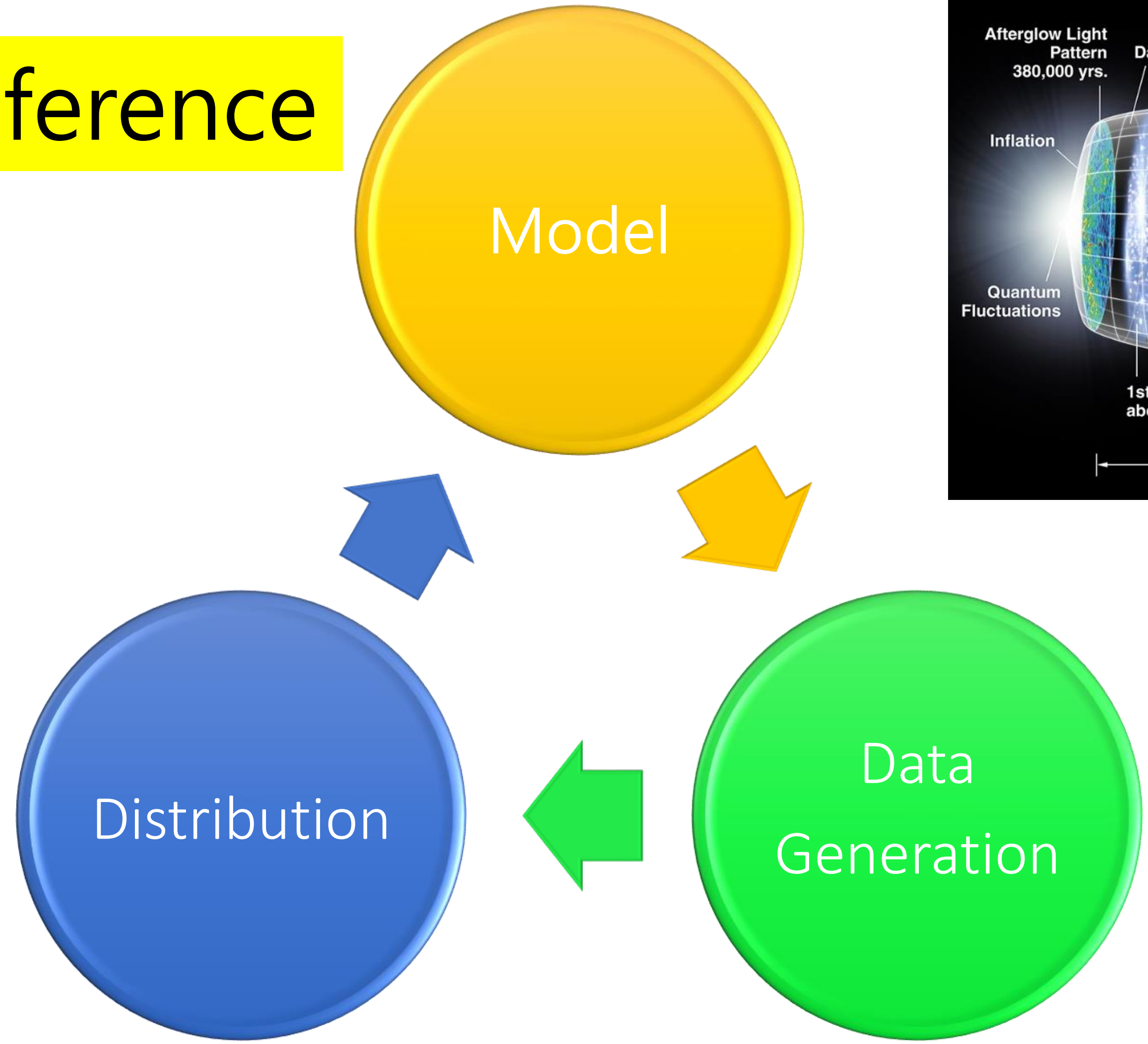


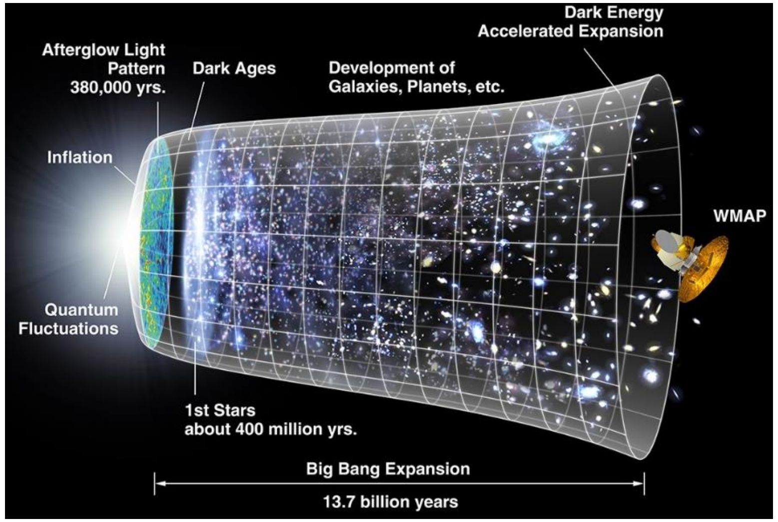


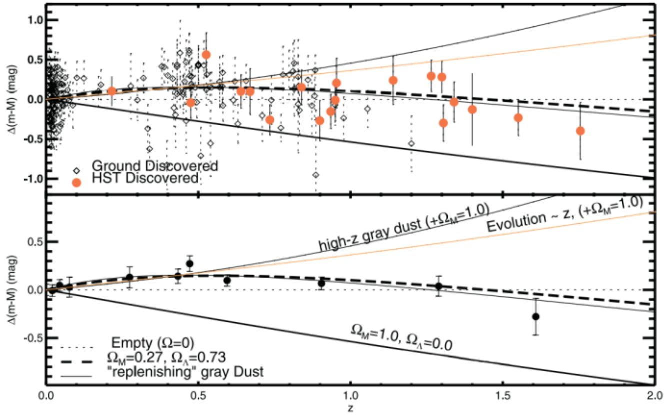
Classical vs. Bayesian Statistical Inference

Bayesian Inference







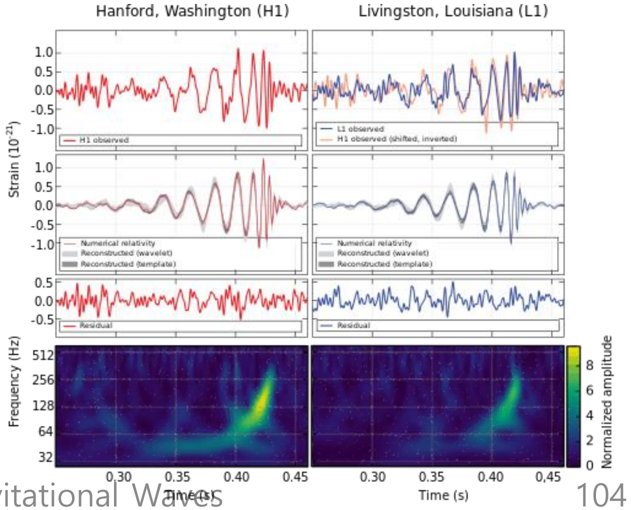


The Likelihood Function

- Data Likelihood represents a quantitative description of our measuring process.

Understand Device → Calculate Probability

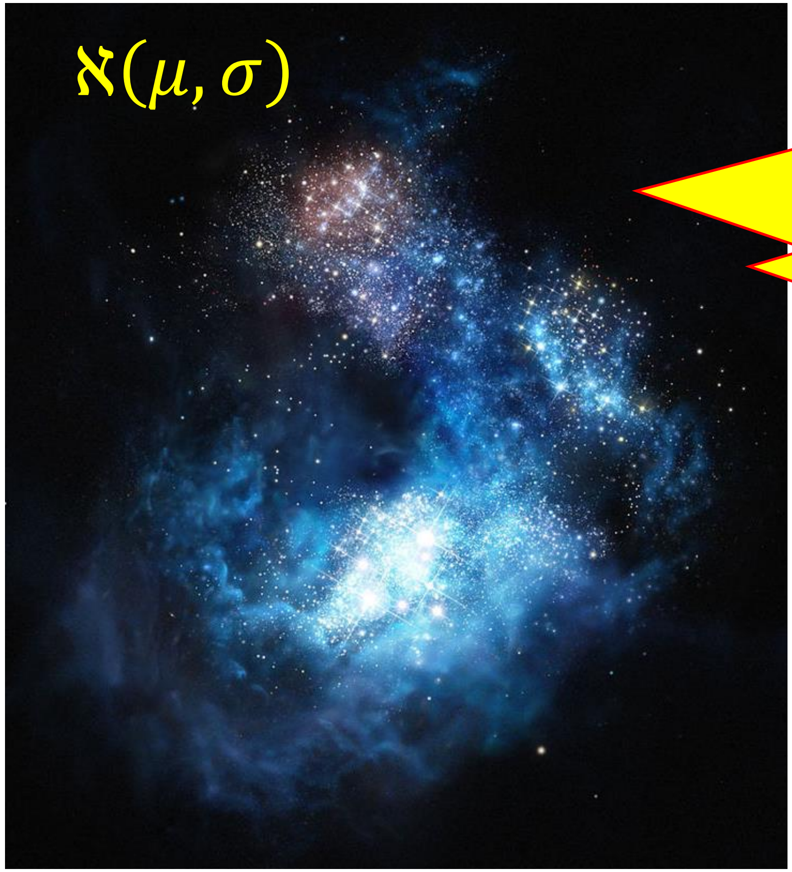
Likelihood

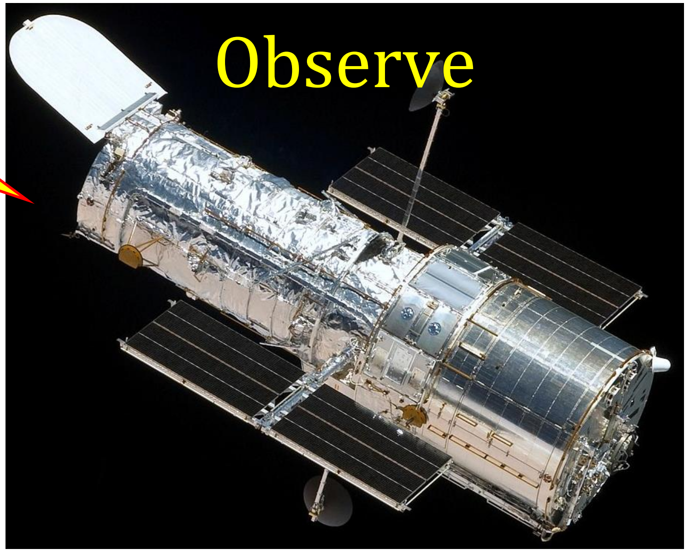


The Likelihood Function

The probability of the data given the model

$$L \equiv p(\{x_i\}|M(\boldsymbol{\theta})) = \prod_{i=1}^{n} p(x_i|M(\boldsymbol{\theta}))$$





$$x_i: \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(\frac{-(x_i-\mu)^2}{2\sigma}\right)$$

Likelihood is a function of x for given model

Likelihood is a function of model parameters for given data

Essence of the Bayesian Idea

- Bayes' rule(theorem)

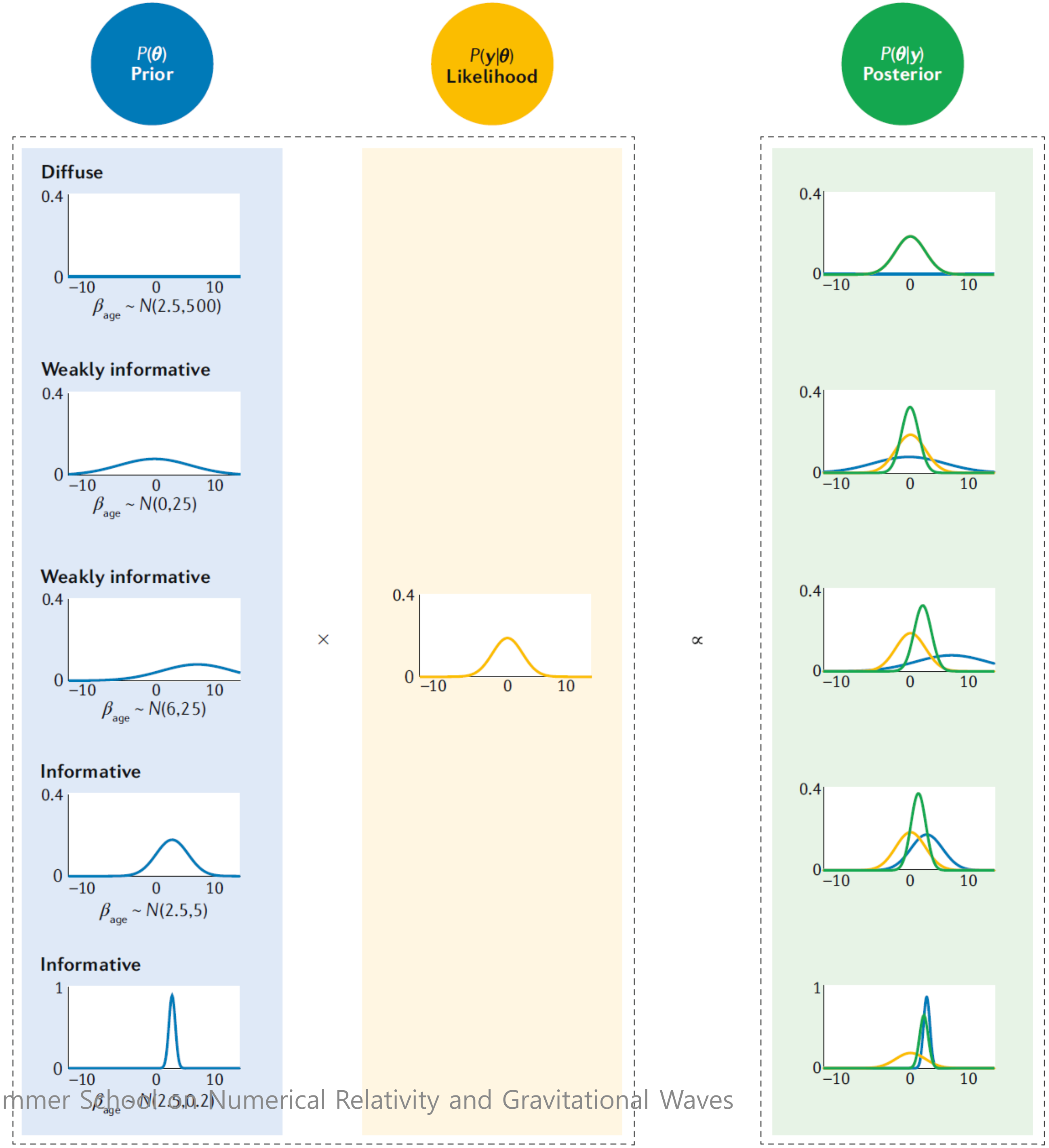
$$p(M|D) = \frac{p(D|M)\,p(M)}{p(D)} \qquad p(M,D) = p(M|D)p(D) = p(D|M)p(M)$$

- Improved belief is the product of initial belief and the probability that initial belief generate the observed data

$$p(M,\boldsymbol{\theta}|D,I) = \frac{p(D|M,\boldsymbol{\theta},I)\,p(M,\boldsymbol{\theta}|I)}{p(D|I)}$$

$$p(M,\boldsymbol{\theta}|I) = p(\boldsymbol{\theta}|M,I)\,p(M|I)$$

How related Prior, Likelihood, Posterior



Summary of the Bayesian statistical inference

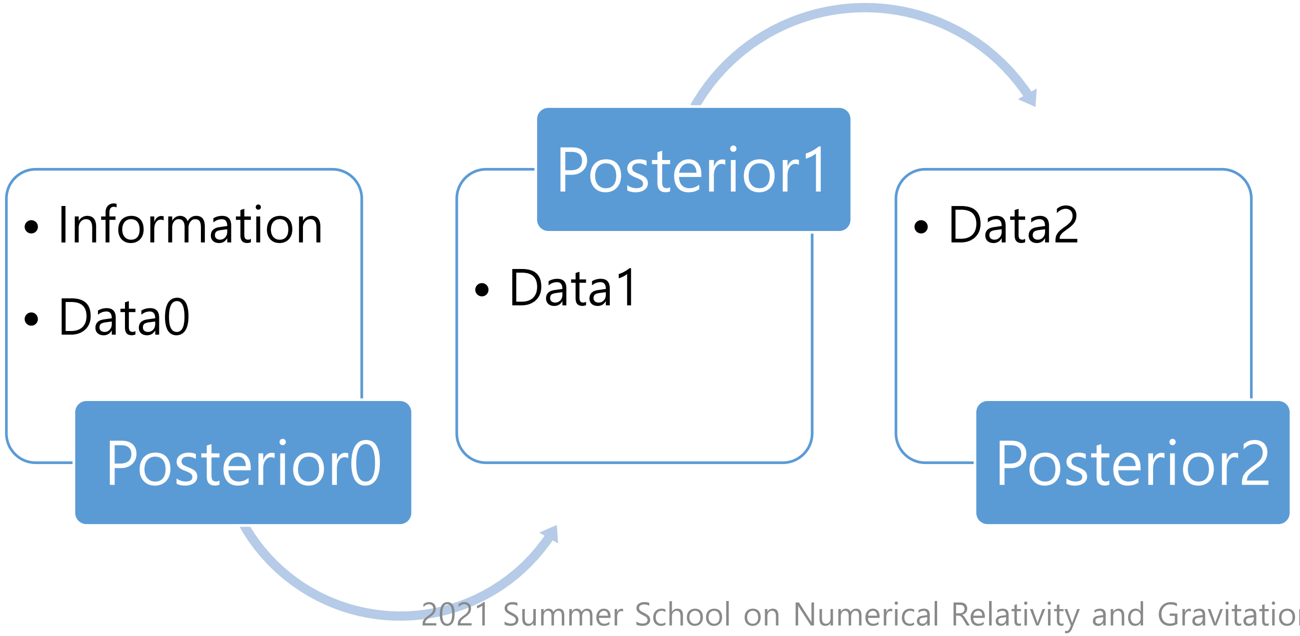
1. Formulation of Likelihood $p(D|M,I)$
2. The choice of prior $p(\vec{\theta}|M,I)$
3. Determination of posterior pdf $p(M|D,I)$
4. Maximum posteriori estimation(MAP)
   posterior mean $\bar{\theta} = \int \theta p(\theta|D) d\theta$
   $p(\theta|D) = \int p\left(\vec{\theta}\middle|D\right) d\overrightarrow{\theta'}$, marginalization over all other parameters
5. Quantification of uncertainty in parameters
6. Hypothesis test about the model or parameters

Comparison of Classical and Bayesian

Classical	Bayesian	Note
$p(D\|M)$	$p(D\|M)$	same
Find maximum $p(D\|M)$ value	Choose prior $p(M)$	
	Determine posterior $p(M\|D)$	
Determine confidence level	Marginalize to determine value	
	Determine confidence level	
Hypothesis test	Hypothesis test	same but different way

Bayesian Priors

- How do we choose prior $p(\vec{\theta}|M, I)$
  - Knowledge extracted from prior measurements
  - Different measurement with a new data
- Informative priors : information based on the other measurement



Parameter Estimation

- ***How probable the parameters for given observation***
- Bayesian Inference[PRD.91.042003]
  - $\wp(\theta|x) = \frac{\wp(\theta,x)}{\wp(x)} = \frac{\wp(x|\theta)\wp(\theta)}{\int \wp(x|\theta)\wp(\theta)d\theta}$
  - $\wp(\theta,x) = \wp(x|\theta)\wp(\theta) = \wp(\theta|x)\wp(x)$
  - θ : unobservable model parameters
  - x : observable data
  - $\wp(\theta,x)$: Joint probability observing data x with model parameter θ
  - $\wp(x|\theta) \propto Likelihood\ function$, probability detecting signal x with θ
- Metropolis-Hasting Algorithm, $10^6 \sim 10^7$ samples
- Extract some independent samples

Parameter Estimation(MCMC)

- Determine posterior $\wp(\theta|x)$ with some methods
- Need to know priors for a given parameter set θ
- $\wp(\theta)$: priors for a given θ

- All physical priors are assumed to be 1(a priori we don’t know)
- Distance prior
- Mass prior

Distance Prior

- Sources are uniformly distributed over space
- Sample position should be generated uniformly over space

1. $dV = r^2 \sin\theta \, dr d\theta d\phi = -r^2 dr d\mu d\phi,\ \mu = \cos\theta$
   - Generate $r \sim [r_{min}, r_{max}], \mu \sim [-1,1], \phi \sim [0, 2\pi]$
   - $\wp(r) = r^2$
2. $dV = -r^2 dr d\mu d\phi = -r^3 d \ln r \, d\mu d\phi,\ \mu = \cos\theta$
   - Generate $\ln r \sim [\ln r_{min}, \ln r_{max}], \mu \sim [-1,1], \phi \sim [0, 2\pi]$
   - $\wp(r) = r^3$

Mass Prior

- Sources are uniformly distributed over m_1, m_2
- Sample masses should be generated uniformly over mass ranges

1. $dm_1 dm_2 = m_c(1+q)^{2/5} q^{-6/5} dm_c dq$
   - Generate $m_c \sim [m_c^{min}, m_c^{max}], q \sim [0,1]$
   - $\wp(m_c, q) = \frac{m_1^2}{m_c} = m_c(1+q)^{2/5} q^{-6/5}$
2. $dm_1 dm_2 = \frac{(m_1+m_2)^2}{(m_1-m_2)\eta^{-3/5}} dm_c d\eta = \frac{m_c\left(1+\sqrt{1-4\eta}\right)}{1-4\eta+\sqrt{1-4\eta}} dm_c d\eta$
   - Generate $m_c \sim [m_c^{min}, m_c^{max}], \eta \sim [0, 0.25]$
   - $\wp(m_c, \eta) = \frac{m_c\left(1+\sqrt{1-4\eta}\right)}{1-4\eta+\sqrt{1-4\eta}}$

Singular at $\eta = 0.25$ or $m_1 = m_2$

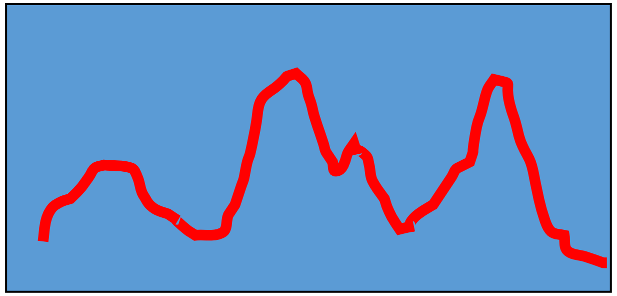
$$m_c = \frac{(m_1 m_2)^{3/5}}{(m_1+m_2)^{1/5}}$$
$$q = \frac{m_2}{m_1}$$
$$\eta = \frac{m_1 m_2}{(m_1+m_2)^2}$$

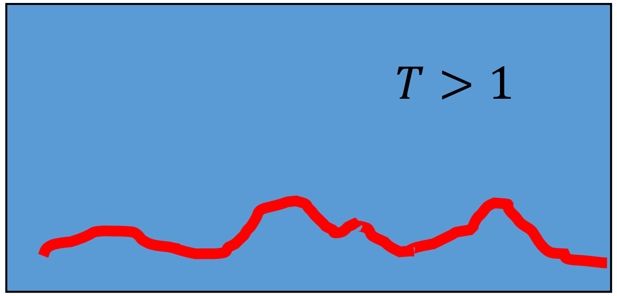
Likelihood Calculation

- Likelihood
  $$\log \mathcal{L} = -\frac{1}{2}\langle d-h|d-h\rangle = -\frac{1}{2}\langle d|d\rangle + \langle d|h\rangle - \frac{1}{2}\langle h|h\rangle$$
- Gaussian noise assumption
- $\wp(x|\theta) = \mathcal{L}(x|\theta)$

Parallel Tempering

- Use few chains with different temperature
- Use likelihood $\mathcal{L}(s|\theta)^{\frac{1}{T}}, \;\; T > 1$
- $T_{max} = \dfrac{(Network\ SNR)^2}{n_{par}}$
- Improve convergence and mixing
- The higher temperature, the smoother distribution





Inner Product

- $\langle a|b\rangle \equiv 4\Re \int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{S(f)} df$
- $\langle a|b\rangle = 2\Re \int_{-\infty}^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{S(|f|)} df = \int_{-\infty}^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)+\tilde{a}^*(f)\tilde{b}(f)}{S(|f|)} df$
- $p_x[x(t)] \propto e^{-\langle x|x\rangle/2}$

- We can consider each frequency bin distributed as
- $p_x[\tilde{x}(f)] = \frac{1}{\sqrt{2\pi}}\sqrt{\frac{4\Delta f}{S_x(f)}}\, e^{-\frac{1}{2}\frac{|\tilde{x}(f)|^2 \Delta f}{S_x(f)/4}}$

$$p_x[x(t)] \propto \exp\left\{-\frac{1}{2}4\int_0^\infty \frac{|\tilde{x}(f)|^2}{S_x} df\right\}$$

$$\Delta f = \frac{1}{T}$$

Likelihood(definition)

- $\wp(d|\theta)$: probability detecting signal d with θ
- Probability of the signal d contains template $h(\theta)$
  - ***If $\mathbf{d}$ contains template $\mathbf{h(\theta)}$, then $\mathbf{d-h}$ is purely noise***
- $\wp(d-h|\theta) = \wp(n|\theta)$
- $p_d[\tilde{d}(f_j) - \tilde{h}(f_j)] = \frac{1}{\sqrt{2\pi}}\sqrt{\frac{4\Delta f}{S_n(f_j)}}e^{-\frac{1}{2}\frac{\left|\tilde{d}(f_j)-\tilde{h}(f_j)\right|^2\Delta f}{S_n(f_j)/4}}$
- $\wp(d|\theta) = \prod_{j=0}^{N-1}\frac{1}{\sqrt{2\pi}}\sqrt{\frac{4\Delta f}{S_n(f_j)}}e^{-\frac{1}{2}\frac{\left|\tilde{d}(f_j)-\tilde{h}(f_j)\right|^2\Delta f}{S_n(f_j)/4}} \propto e^{-\langle d-h|d-h\rangle/2}$

Likelihood(properly normalized)

- $\wp(d|\theta) = \prod_{j=0}^{N-1} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{4\Delta f}{S_n(f_j)}} e^{-\frac{1}{2}\frac{\left|\tilde{d}\left(f_j\right)-\tilde{h}\left(f_j\right)\right|^2 \Delta f}{S_n(f_j)/4}}$
- $\wp(d|\theta) = \left(\prod_{j=0}^{N-1} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{4\Delta f}{S_n(f_j)}}\right) e^{-\frac{1}{2}\sum_{j=0}^{N-1}\frac{\left|\tilde{d}\left(f_j\right)-\tilde{h}\left(f_j\right)\right|^2 \Delta f}{S_n(f_j)/4}}$
- $\ln \wp(d|\theta) = -\frac{1}{2}\sum_{j=0}^{N-1} \frac{\left|\tilde{d}\left(f_j\right)-\tilde{h}\left(f_j\right)\right|^2 \Delta f}{S_n\left(f_j\right)/4} - \frac{1}{2}\sum_{j=0}^{N-1} \ln \frac{2\pi S_n\left(f_j\right)}{4\Delta f}$
- $\ln \wp(d|\theta) = \sum_{j=0}^{N-1}\left[-\frac{2\left|\tilde{d}\left(f_j\right)-\tilde{h}\left(f_j\right)\right|^2 \Delta f}{S_n\left(f_j\right)} - \frac{1}{2}\ln \frac{\pi S_n\left(f_j\right)}{2\Delta f}\right]$

Likelihood Calculation (used)

- Likelihood

$$\sum_{j=0}^{N-1}\left[-\frac{2\left|\tilde{d}(f_j)-\tilde{h}(f_j)\right|^2\Delta f}{S_n(f_j)}\right]$$

$$\log\mathcal{L} = -\frac{1}{2}\langle d-h|d-h\rangle = -\frac{1}{2}\langle d|d\rangle + \langle d|h\rangle - \frac{1}{2}\langle h|h\rangle$$

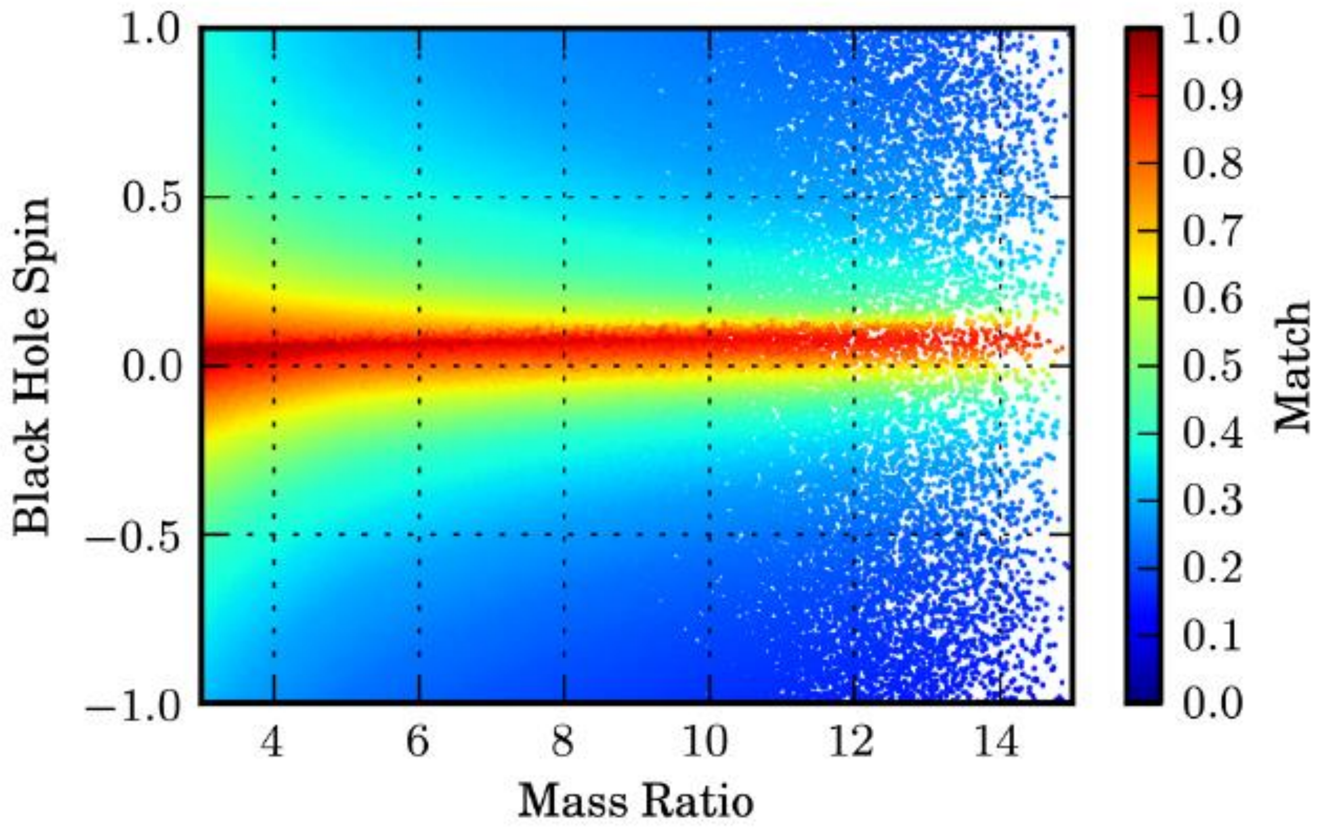
- Gaussian noise assumption
- $\wp(d|\theta) = \mathcal{L}(d|\theta)$

$$\langle d-h|d-h\rangle = 4\int_0^\infty \frac{(d-h)^*(d-h)}{S_n(f)}df \simeq 4\int_{f_{\mathrm{Low}}}^{f_{\mathrm{High}}} \frac{(d-h)^*(d-h)}{S_n(f)}df$$

Overlap

PRD88, 123039(2013)

- $\mathcal{O}(h_1, h_2) = \dfrac{\langle h_1|h_2\rangle}{\sqrt{\langle h_1|h_1\rangle\langle h_2|h_2\rangle}}$
- $SNR = \sqrt{\langle h|h\rangle}$



Markov Chain Monte Carlo

Markov Chain Monte Carlo

Markov Process

- Definition
  - Number of state at any time is finite
  - Next sate depends only on the current state
  - Any state could be arrived from any state
  - System does not have deterministic cycle
  - Transition kernel $P(X_t \to X_{t+1})$ constant
- Ergodicity
  - Any Markov Process converge to a unique statistical equilibrium from any state

Detailed Balance

- At statistical equilibrium, probability of state should not change
  - $\pi(X_t)P(X_t \to X_{t+1}) = \pi(X_{t+1})P(X_{t+1} \to X_t)$
  - $\frac{P(X_t \to X_{t+1})}{P(X_{t+1} \to X_t)} = \frac{\pi(X_{t+1})}{\pi(X_t)}$
- Practical implementation
  - Proposal : $q(X_t \to X_{t+1})$
  - Acceptance : $\alpha(X_t \to X_{t+1})$
  - $P(X_t \to X_{t+1}) = q(X_t \to X_{t+1})\alpha(X_t \to X_{t+1})$

Detailed Balance

- Detailed balance condition acquired if
  - $\frac{P(X_t \to X_{t+1})}{P(X_{t+1} \to X_t)} = \frac{q(X_t \to X_{t+1})\alpha(X_t \to X_{t+1})}{q(X_{t+1} \to X_t)\alpha(X_{t+1} \to X_t)} = \frac{\pi(X_{t+1})}{\pi(X_t)}$
  - $\frac{\alpha(X_t \to X_{t+1})}{\alpha(X_{t+1} \to X_t)} = \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}$
- Choose next state with probability
  - $\alpha(X_t \to X_{t+1}) = \min\left(1, \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}\right)$

Detailed Balance Proof

- $\pi(X_{t+1})q(X_{t+1} \to X_t) > \pi(X_t)q(X_t \to X_{t+1})$ case
  - $\alpha(X_t \to X_{t+1}) = \min\left(1, \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}\right) = 1$
  - $\alpha(X_{t+1} \to X_t) = \frac{\pi(X_t)q(X_t \to X_{t+1})}{\pi(X_{t+1})q(X_{t+1} \to X_t)}$
  - $\frac{\alpha(X_t \to X_{t+1})}{\alpha(X_{t+1} \to X_t)} = \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}$

Detailed Balance Proof

- $\pi(X_{t+1})q(X_{t+1} \to X_t) < \pi(X_t)q(X_t \to X_{t+1})$ case
  - $\alpha(X_t \to X_{t+1}) = \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}$
  - $\alpha(X_{t+1} \to X_t) = \min\left(1, \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}\right) = 1$
  - $\frac{\alpha(X_t \to X_{t+1})}{\alpha(X_{t+1} \to X_t)} = \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}$

[(68) (ML 18.6) Detailed balance (a.k.a. Reversibility) - YouTube]()

Metropolis Algorithm

- $\alpha(X_t \to X_{t+1}) = \min\left(1, \frac{\pi(X_{t+1})}{\pi(X_t)}\right)$: acceptance rate
- Metropolis Algorithm is for the case of $q(X_t \to X_{t+1}) = q(X_{t+1} \to X_t)$
- $q(X_t \to X_{t+1})$ Gaussian distribution centered at X_t

Metropolis-Hasting Algorithm

- $\alpha(X_t \to X_{t+1}) = \min\left(1, \frac{\pi(X_{t+1})q(X_{t+1} \to X_t)}{\pi(X_t)q(X_t \to X_{t+1})}\right)$: acceptance rate
- One can choose arbitrary appropriate $q(X_t \to X_{t+1})$ for the target problem

MCMCs

Table 1 | **An overview of MCMC-based and non-MCMC-based sampling techniques**

Approach	Short description
MCMC-based methods	
Metropolis–Hastings	An algorithm used for obtaining random samples from a probability distribution. Uses a general proposal distribution, with an associated accept/reject step for the proposed parameter value(s)[85,86]
Reversible jump MCMC	An extension of the Metropolis–Hastings algorithm. Permits simulation of trans-dimensional moves within parameter space[35,225]
Hamiltonian Monte Carlo	A Metropolis–Hastings algorithm based on Hamiltonian dynamics[87]. This algorithm is useful if direct sampling is difficult, if the sample size is small or when autocorrelation is high. The algorithm avoids the random walk of Metropolis–Hastings and sensitivity by taking a series of steps informed by first-order gradient information. The No-U-Turn Sampler[226] is an extension and is often faster because it often avoids the need for tuning the model
Gibbs sampler	A Metropolis–Hastings algorithm where the proposal distribution is the corresponding posterior conditional distribution, with an associated acceptance probability of 1 (REF.[84])
Particle MCMC	A combined sequential Monte Carlo algorithm and MCMC used when the likelihood is analytically intractable[177]
Evolutionary Monte Carlo	An MCMC algorithm that incorporates features of genetic algorithms and simulated annealing[227]. It allows the Markov chain to effectively and efficiently explore the parameter space and avoid getting trapped at local modes of the posterior distribution. It is particularly useful when the target distribution function is high-dimensional or multimodal
Non-MCMC-based methods	
Sequential Monte Carlo	An algorithm based on multiple importance sampling steps for each observed data point. Often used for online or real-time processing of data arrivals[228]
Approximate Bayesian computation	An approximate approach, typically used when the likelihood function is analytically intractable or very computationally expensive[229]
Integrated nested Laplace approximations	An approximate approach developed for the large class of latent Gaussian models, which includes generalized additive spline models, Gaussian Markov processes and random fields[230]
Variational Bayes	Variational inference describes a technique to approximate posterior distributions via simpler approximating distributions. The popular mean-field approximation assigns an approximating variational distribution to each parameter independently. Gradient descent is then used to optimize the variational parameters to minimize a loss function known as the evidence lower bound[99]

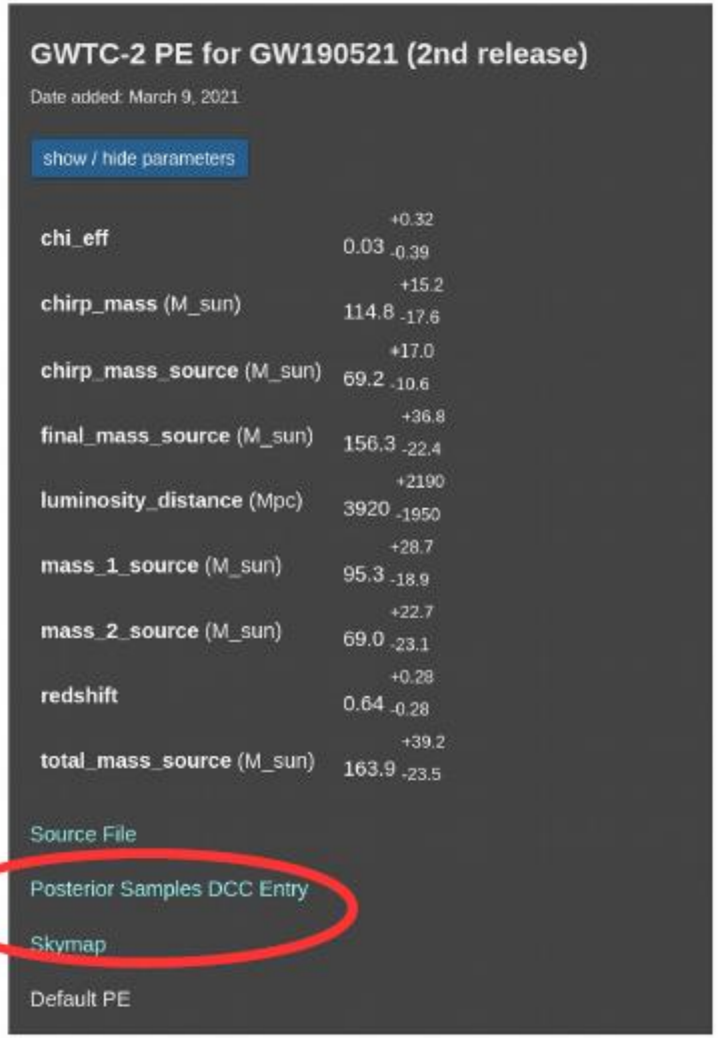
MCMC, Markov chain Monte Carlo.

Assumptions on PE Likelihoods

[Creighton Chap.7, PRD.91.042003, PRD.85.122006]

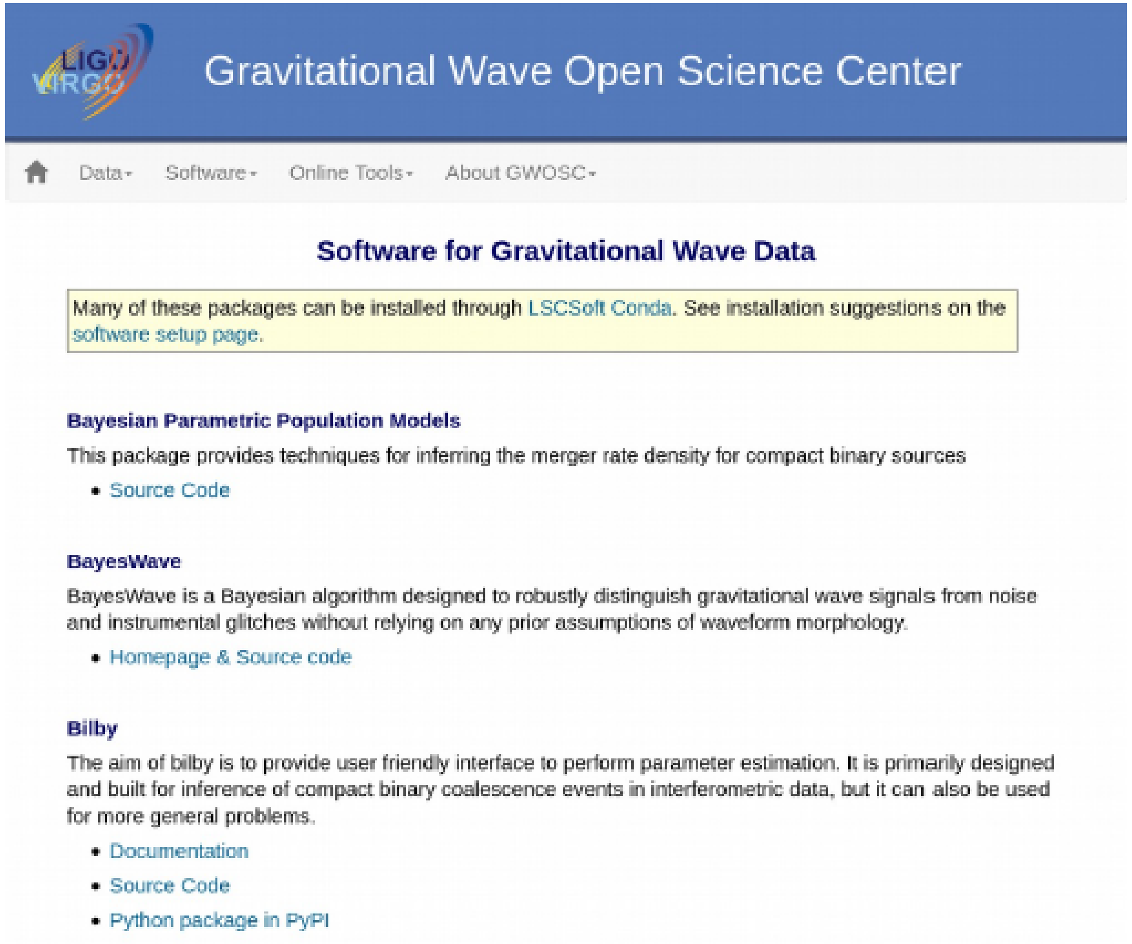
- Noise is stationary and Gaussian
- No correlation between time bins
- No correlation between frequency bins
- Residual signal should be noise only for matched case

Posterior samples can be downloaded from gw-openscience.org



Resources

https://www.gw-openscience.org/software/



Gravitational Wave Open Science Center

Data▾ Software▾ Online Tools▾ About GWOSC▾

Software for Gravitational Wave Data

Many of these packages can be installed through LSCSoft Conda. See installation suggestions on the software setup page.

Bayesian Parametric Population Models

This package provides techniques for inferring the merger rate density for compact binary sources

- Source Code

BayesWave

BayesWave is a Bayesian algorithm designed to robustly distinguish gravitational wave signals from noise and instrumental glitches without relying on any prior assumptions of waveform morphology.

- Homepage & Source code

Bilby

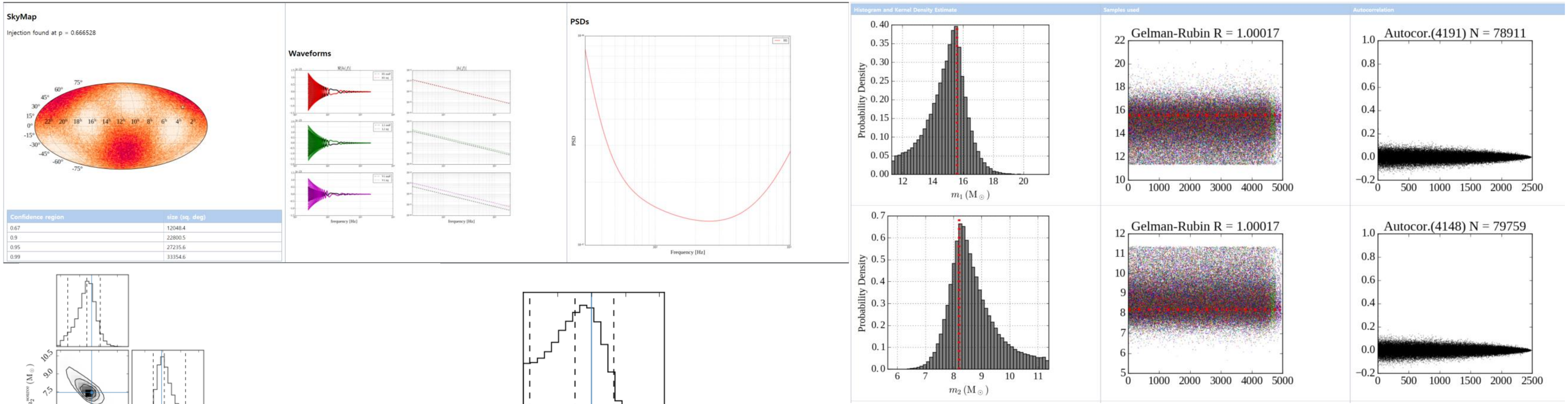
The aim of bilby is to provide user friendly interface to perform parameter estimation. It is primarily designed and built for inference of compact binary coalescence events in interferometric data, but it can also be used for more general problems.

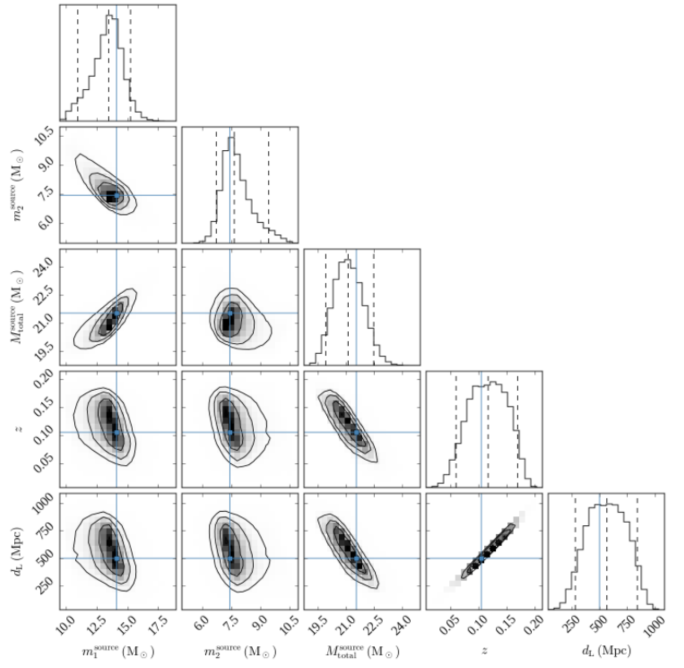
- Documentation
- Source Code
- Python package in PyPI

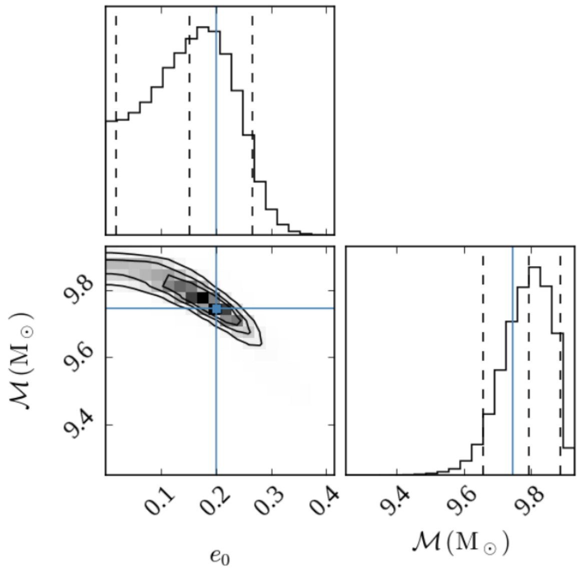
Software packages for Bayesian inference applied to compact binary coalescences

- LALInference – https://git.ligo.org/lscsoft/lalsuite
- Bilby – https://git.ligo.org/lscsoft/bilby
- Bayesian Parametric Population Models – https://git.ligo.org/daniel.wysocki/bayesian-parametric-population-models
- Bayeswave – https://git.ligo.org/lscsoft/bayeswave

Sample Output of PE



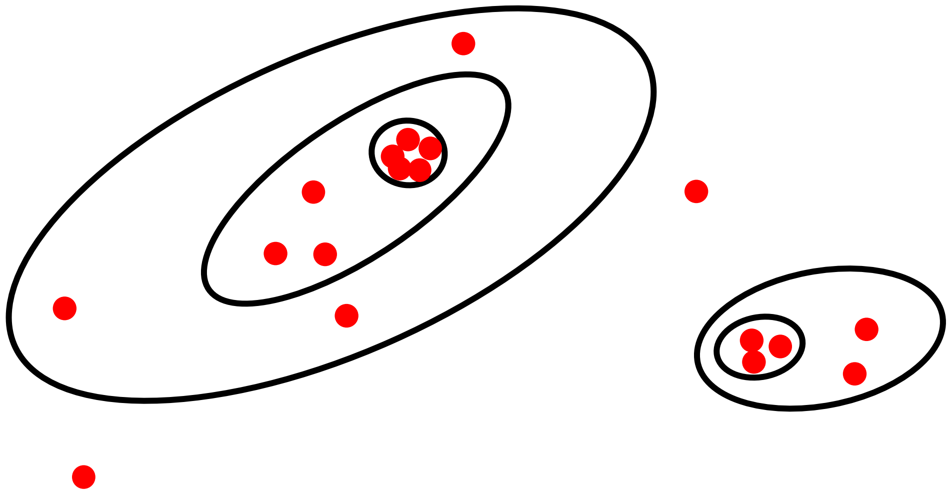


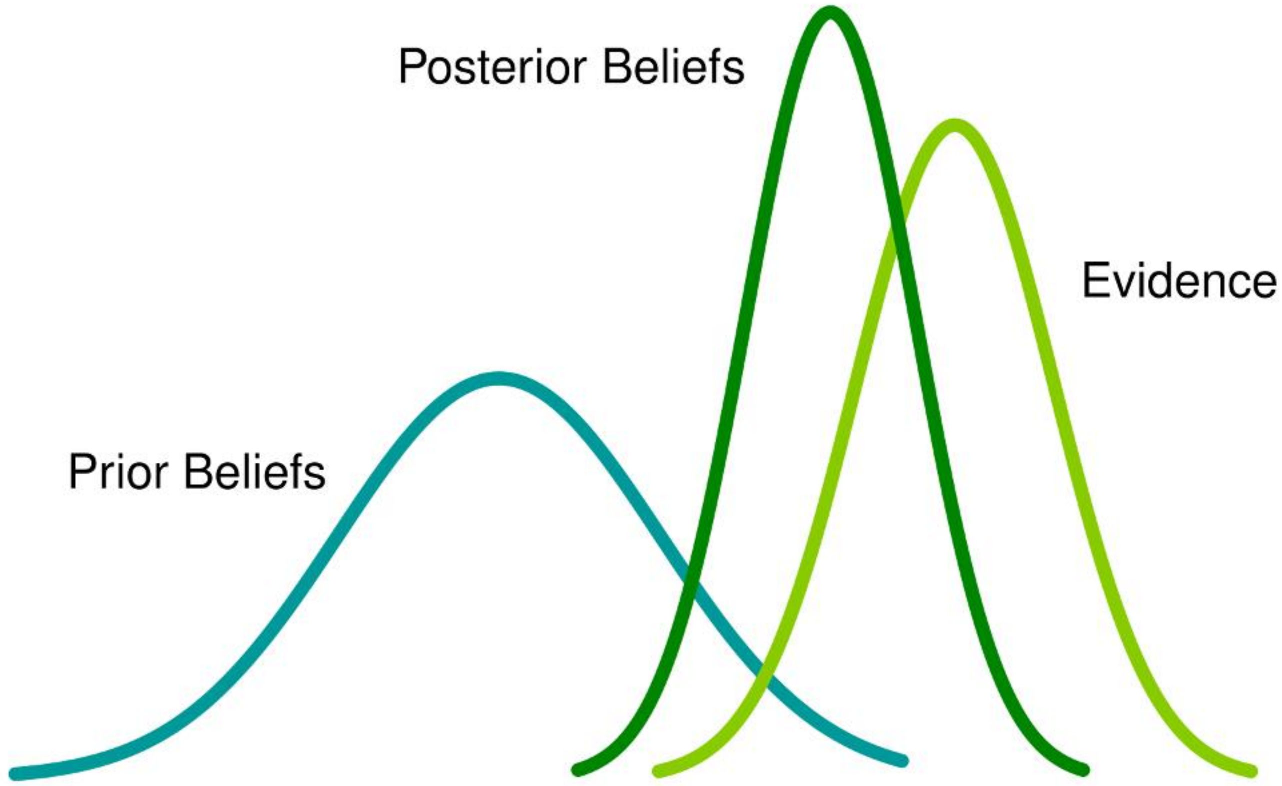


Essence of the Parameter Estimation

- **Generate enough independent samples following posterior**

$$p(M|D) = \frac{p(D|M)\,p(M)}{p(D)}$$

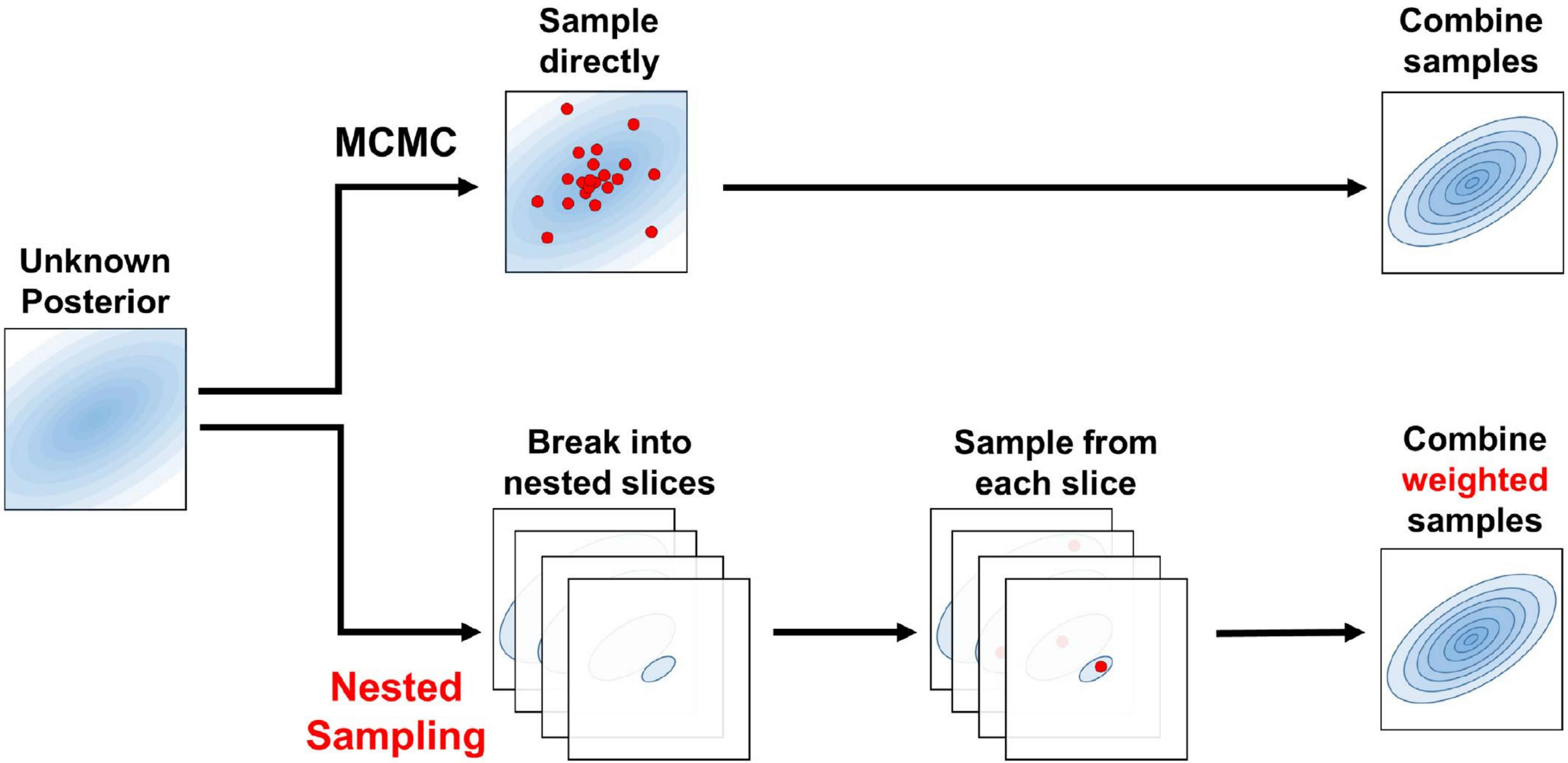




Various PE methods

Nested Sampling

Nested sampling



Nested sampling algorithm(static)

Algorithm 1: Static Nested Sampling

```
// Initialize live points.
```
Draw K "live" points$\{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K\}$ from the prior $\pi(\boldsymbol{\Theta})$.
```
// Main sampling loop.
```
while stopping criterion not met **do**

- Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.
- Add the kth live point $\boldsymbol{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of "dead" points.
- Sample a new point $\boldsymbol{\Theta}'$ from the prior subject to the constraint$\mathcal{L}(\boldsymbol{\Theta}') \geq \mathcal{L}^{\min}$.
- Replace $\boldsymbol{\Theta}_k$ with $\boldsymbol{\Theta}'$.
- `// Check whether to stop.`
- Evaluate stopping criterion.

end
```
// Add final live points.
```
while $K > 0$ **do**

- Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.
- Add the kth live point $\boldsymbol{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of "dead" points.
- Remove $\boldsymbol{\Theta}_k$ from the set of live points.
- Set $K = K - 1$.

end

Nested sampling algorithm(Dynamic)

Algorithm 2: Dynamic Nested Sampling

```
// Initialize first set of live points.
```
Draw K "live" points$\{\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_K\}$ from the prior $\pi(\mathbf{\Theta})$.
```
// Main sampling loop.
```
Set $\mathcal{L}^{\min} = 0$ and $K_0 = K$.

while stopping criterion not met **do**

  `// Get current number of live points.`

  Compute the previous number of live points K and thecurrent number of live points K'.

  if $K' \geq K$ **then**

    `// Add in new live points.`

    while $K' > K$ **do**

      Sample a new point $\mathbf{\Theta}'$ from the prior subject to the constraint$\mathcal{L}(\mathbf{\Theta}') \geq \mathcal{L}^{\min}$.

      Add $\mathbf{\Theta}'$ to the set of live points.

      Set $K = K + 1$.

    end

    `// Replace worst live point.`

    Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of K live points.

    Add the kth live point $\mathbf{\Theta}_k$associated with $\mathcal{L}^{\min}$to a list of "dead" points.

    Replace $\mathbf{\Theta}_k$ with $\mathbf{\Theta}'$.

  else

    `// Iteratively remove live points.`

    while $K' < K$ **do**

      Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of $K = K'$ live points.

      Add the kth live point $\mathbf{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of "dead" points.

      Remove $\mathbf{\Theta}_k$ from the set of live points.

      Set $K = K - 1$.

    end

  end

  `// Check whether to stop.`

  Evaluate stopping criterion.

end

`// Add final live points.`

while there are live points remaining **do**

  Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.

  Add the kth live point $\mathbf{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of "dead" points.

  Remove $\mathbf{\Theta}_k$ from the set of live points.

end

Bilby

Bilby(a user-friendly Bayesian inference library)

- [Welcome to bilby's documentation! — bilby 1.1 documentation (ligo.org)]()
- Arxiv:1811.02042
- Arxiv:2006.00714
- Anaconda environment

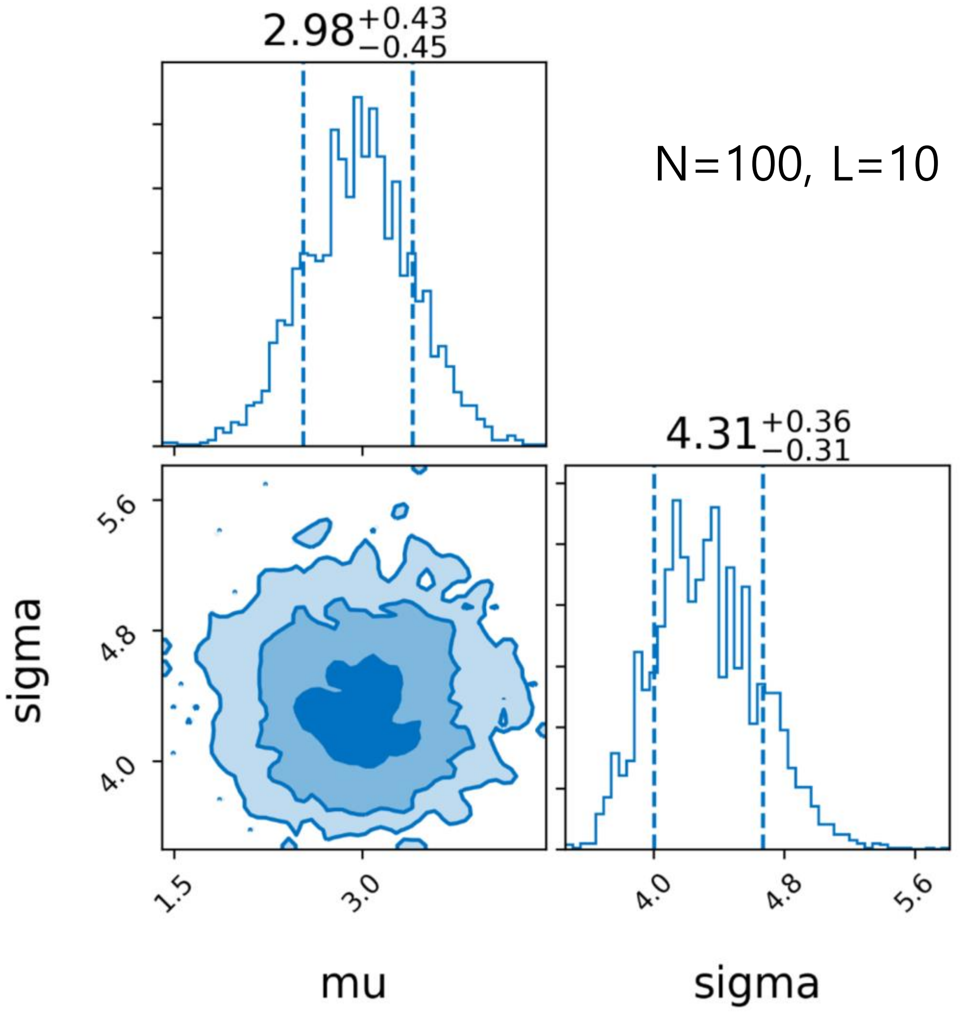
Example Gaussian

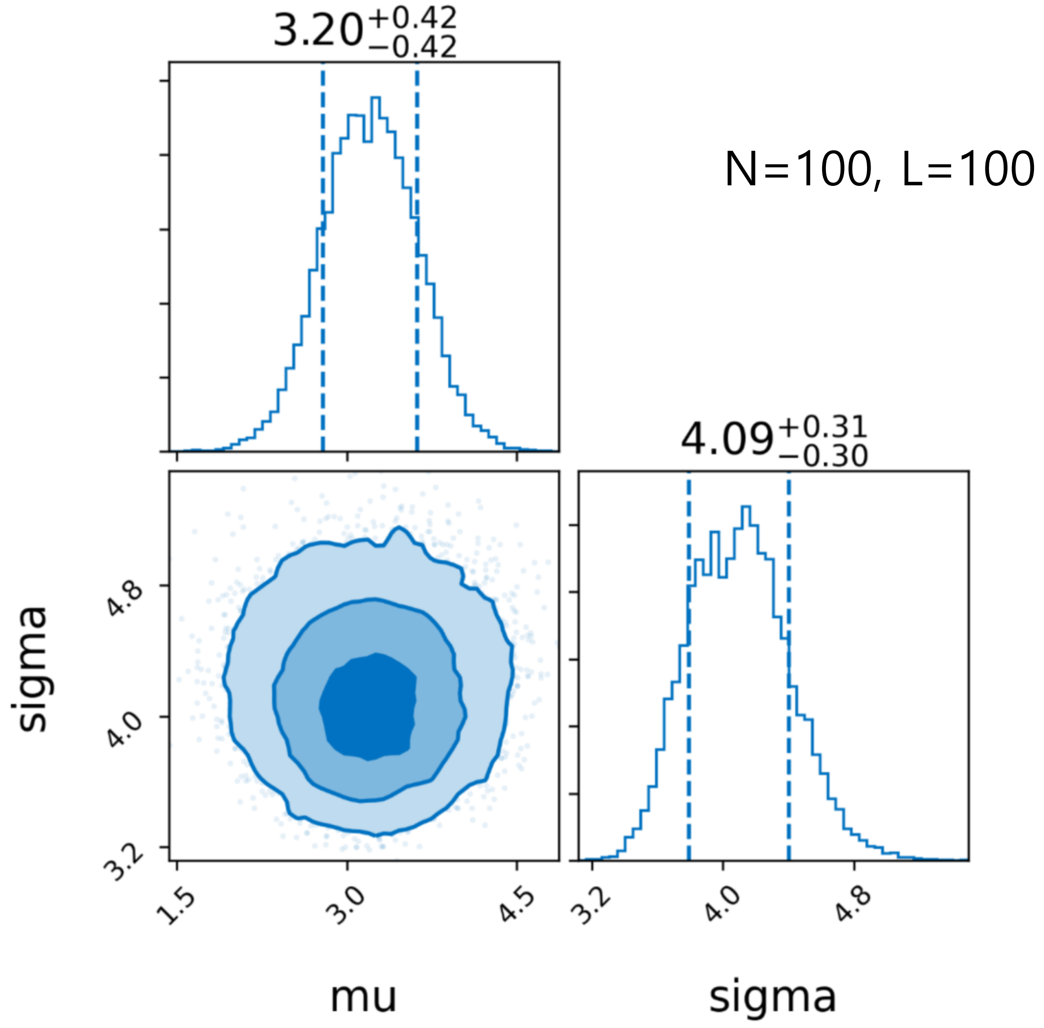
```python
#!/usr/bin/env python3
"""
An example of how to use bilby to perform parameter estimation for
non-gravitational wave data consisting of a Gaussian with a mean and variance
"""
import bilby
import numpy as np

# A few simple setup steps
label = 'gaussian_example'
outdir = 'outdir'

# Here is minimum requirement for a Likelihood class to run with bilby. In this
# case, we setup a GaussianLikelihood, which needs to have a log_likelihood
# method. Note, in this case we will NOT make use of the `bilby`
# waveform_generator to make the signal.

# Making simulated data: in this case, we consider just a Gaussian

data = np.random.normal(3, 4, 100)


class SimpleGaussianLikelihood(bilby.Likelihood):
    def __init__(self, data):
        """
        A very simple Gaussian likelihood

        Parameters
        ----------
        data: array_like
            The data to analyse
        """
        super().__init__(parameters={'mu': None, 'sigma': None})
        self.data = data
        self.N = len(data)
```

- Data : $N(3,4)$

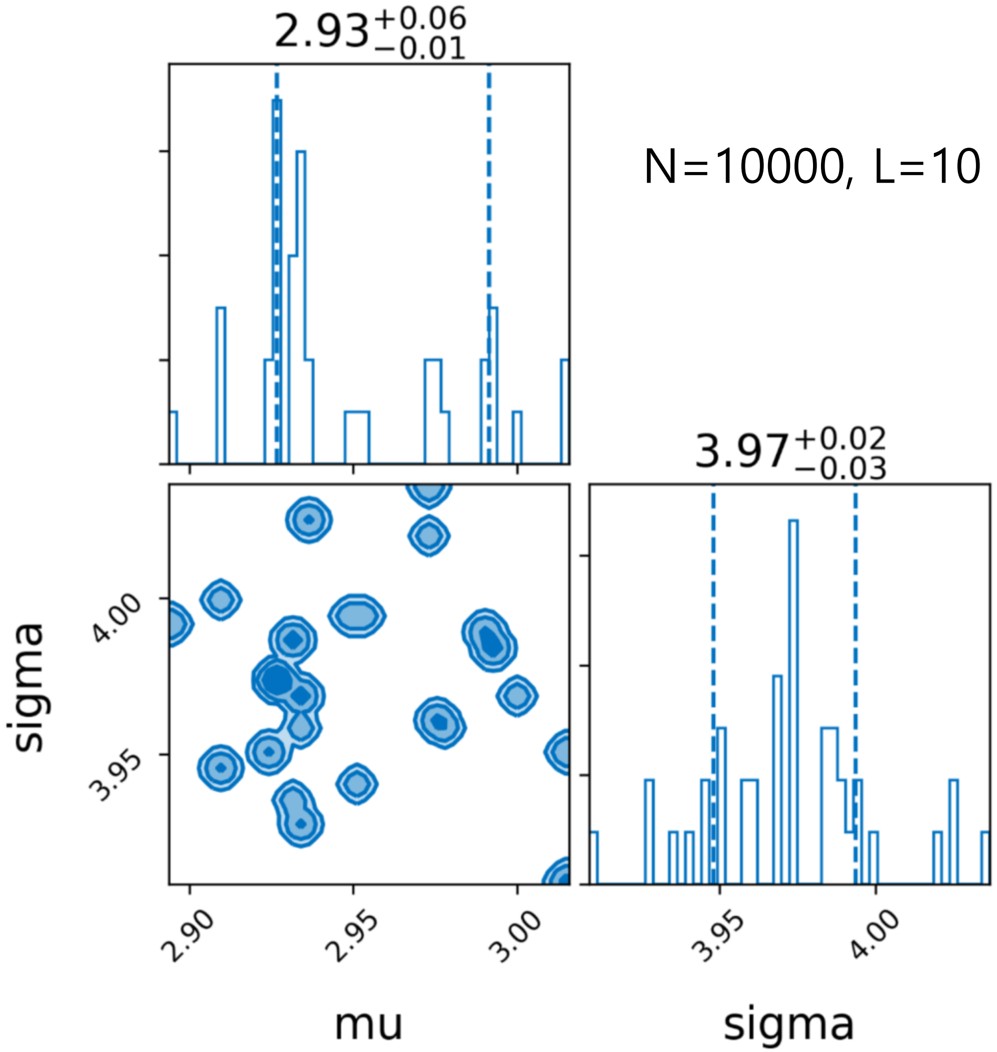
```python
    def log_likelihood(self):
        mu = self.parameters['mu']
        sigma = self.parameters['sigma']
        res = self.data - mu
        return -0.5 * (np.sum((res / sigma)**2) +
                       self.N * np.log(2 * np.pi * sigma**2))


likelihood = SimpleGaussianLikelihood(data)
priors = dict(mu=bilby.core.prior.Uniform(0, 5, 'mu'),
              sigma=bilby.core.prior.Uniform(0, 10, 'sigma'))

# And run sampler
result = bilby.run_sampler(
    likelihood=likelihood, priors=priors, sampler='dnest4', npoints=50000,
    walks=100, outdir=outdir, label=label)
result.plot_corner()
```

Example Gaussian

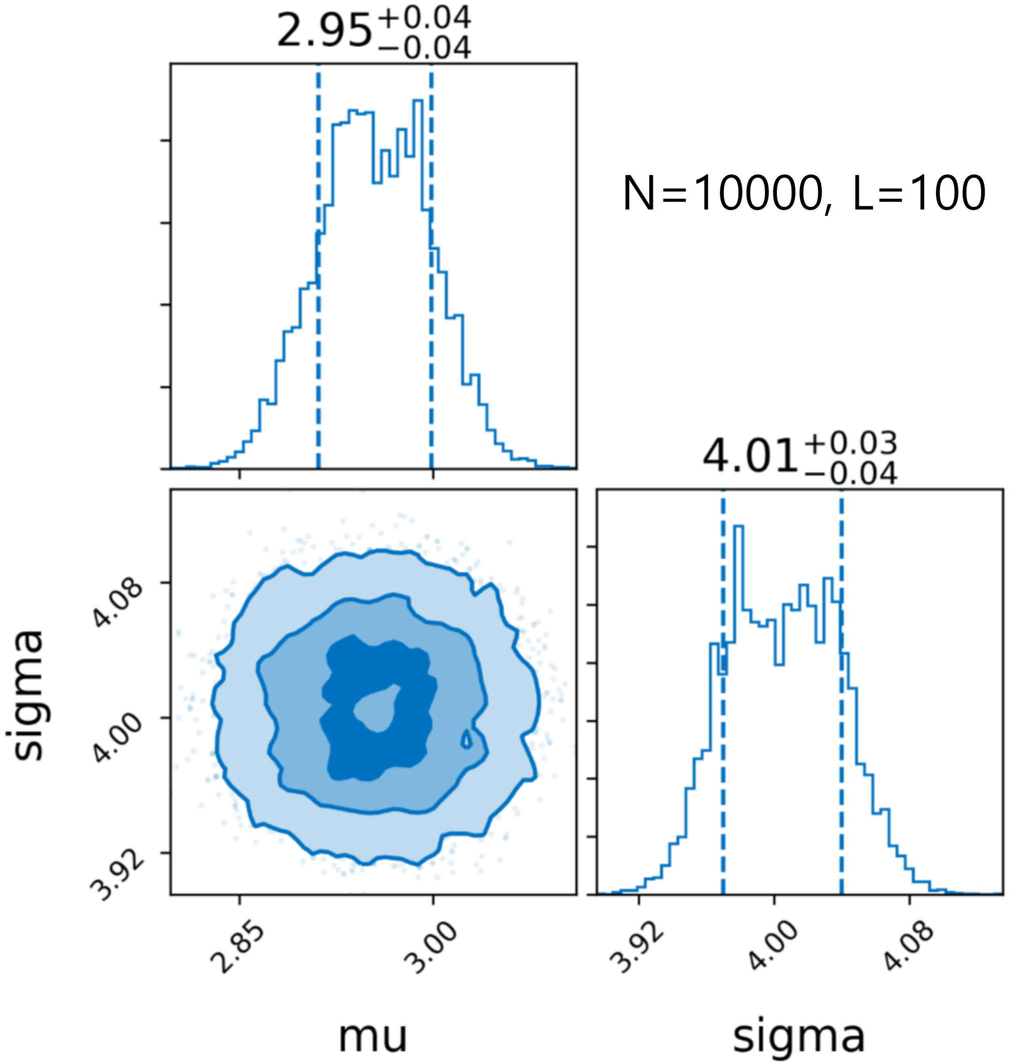




Example Gaussian



N=10000, L=10



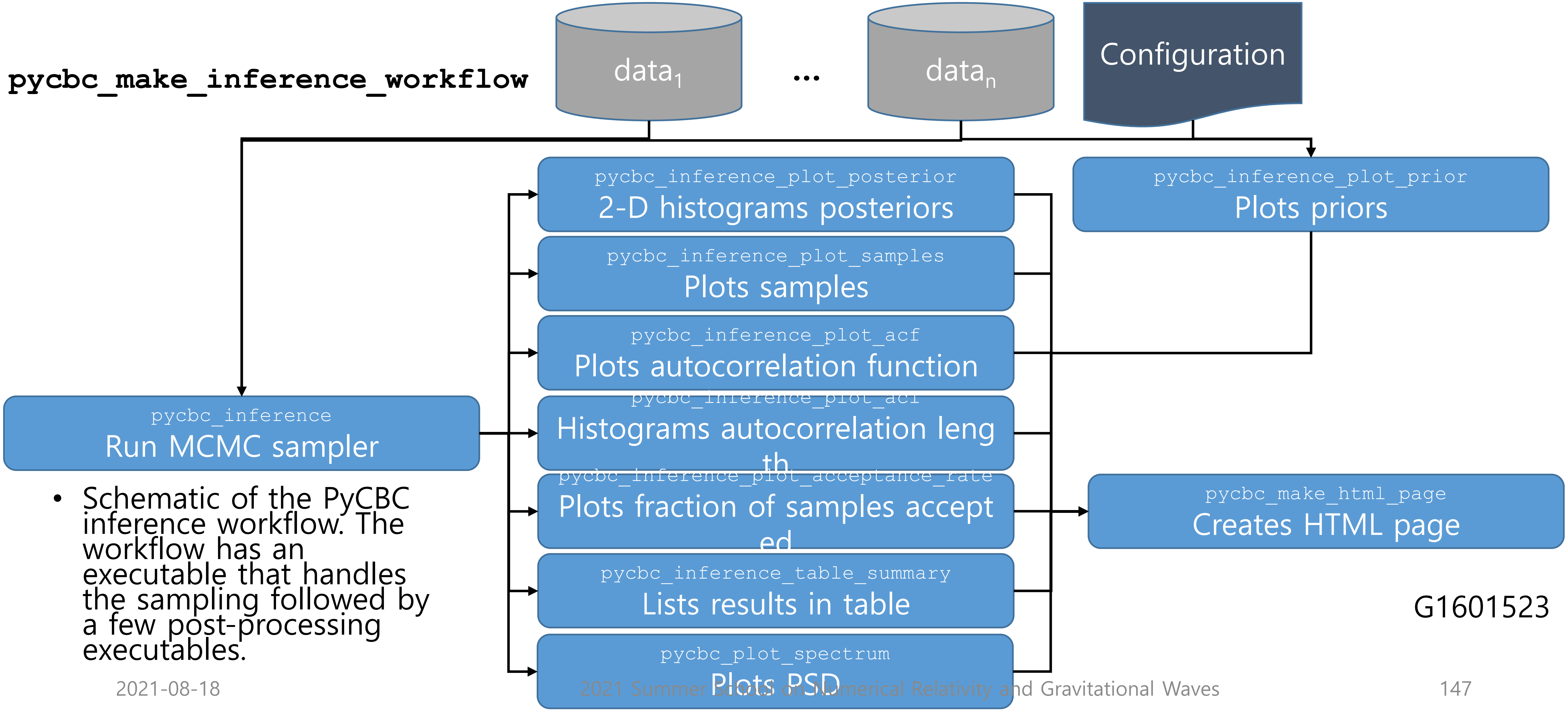
N=10000, L=100

PyCBC Inference

- Python toolkit for CBC analysis (https://github.com/ligo-cbc/pycbc)
  - Workflows
  - Waveform generations
  - PSD estimation
  - Matched filtering
  - Offline PyCBC coincidence search

PyCBC is a software package used to explore astrophysical sources of gravitational waves. It contains algorithms to analyze gravitational-wave data, detect coalescing compact binaries, and make bayesian inferences from gravitational-wave data. PyCBC was used in the first direct detection of gravitational waves and is used in flagship analyses of LIGO and Virgo data.

Workflow topology

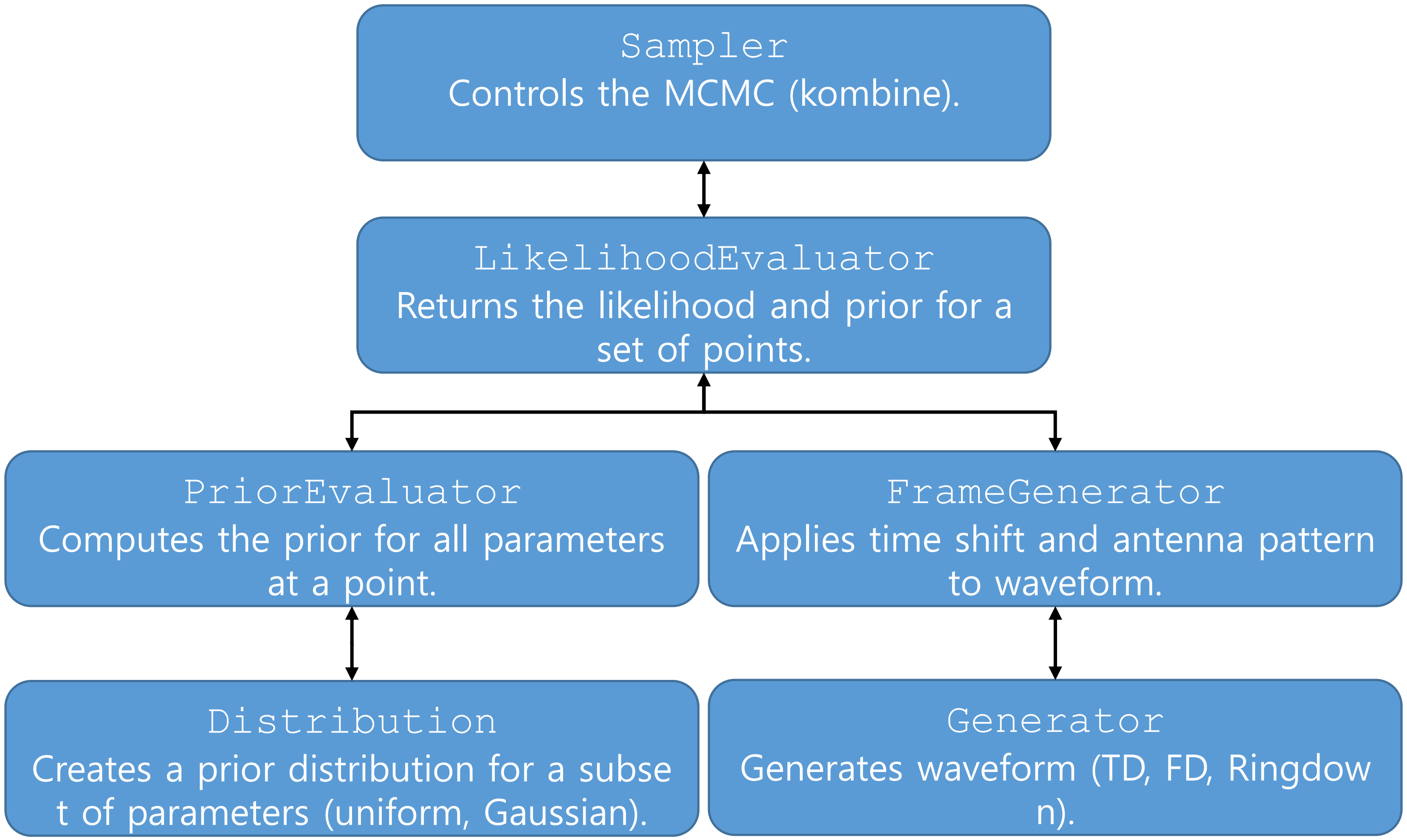


- Schematic of the PyCBC inference workflow. The workflow has an executable that handles the sampling followed by a few post-processing executables.

G1601523

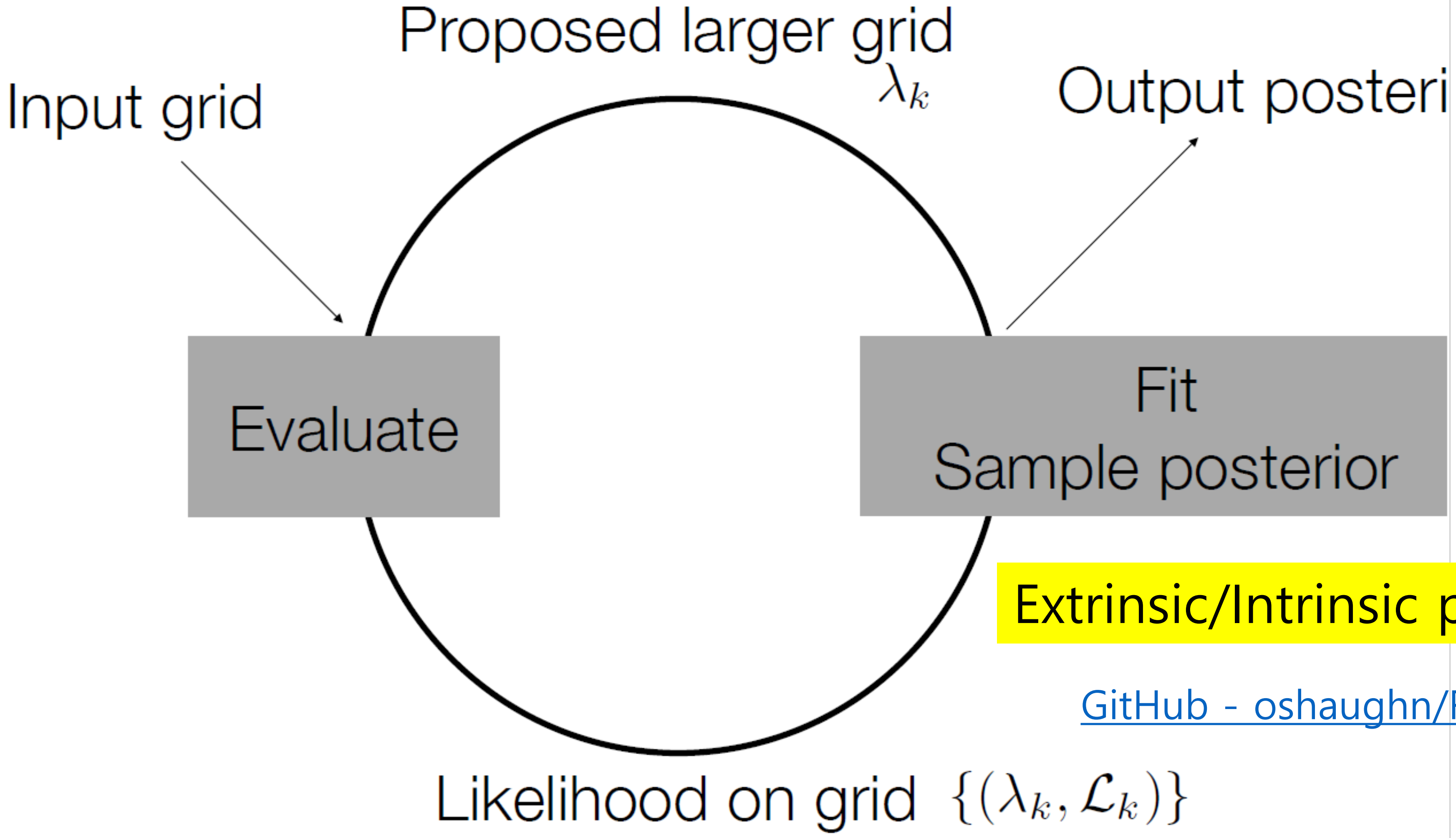
Sampler, waveform, and prior choices

- Samplers:
  - Kombine (MCMC)
- Waveform generators:
  - Time-domain CBC
  - Frequency-domain CBC
  - Ringdown
- Priors:
  - Uniform
  - Uniform angle
  - Uniform sky position
  - Sine and cosine
  - Gaussian
- The graphic on the right shows the PyCBC inference classes for sampling.



Rapid Iterative Fitting(RIFT) for inference

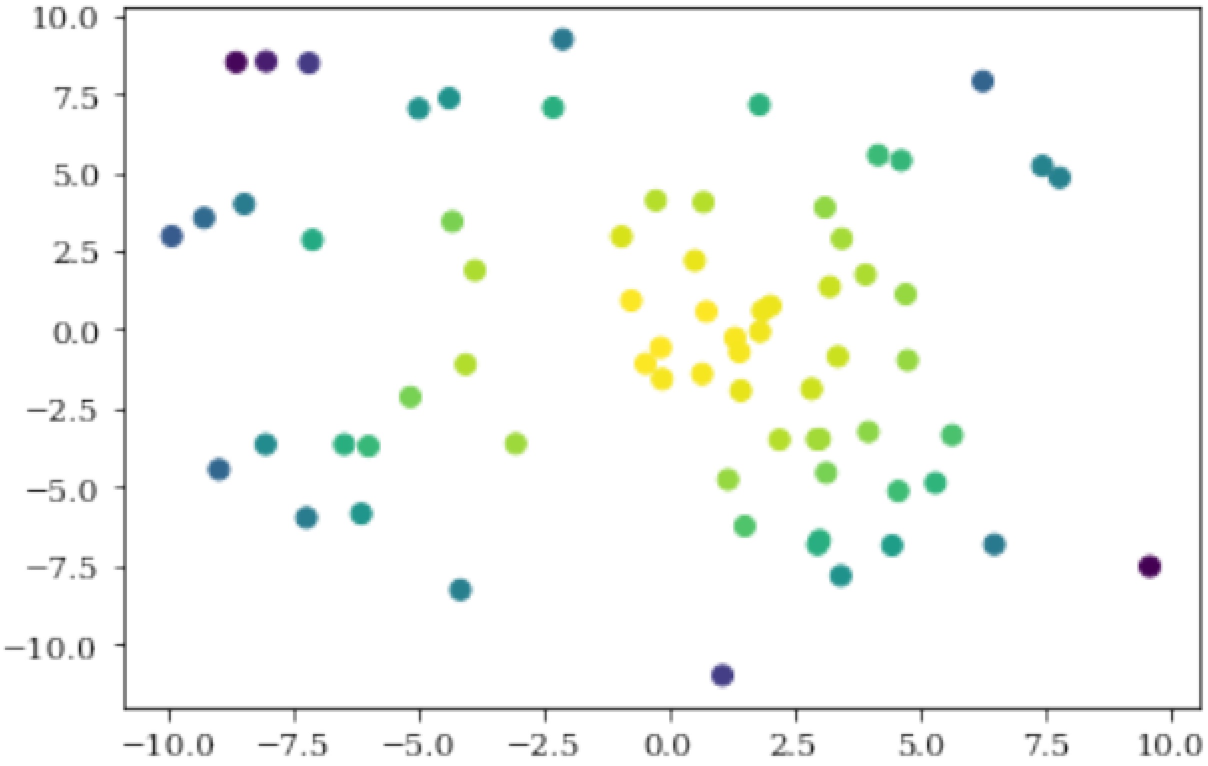
G2100564

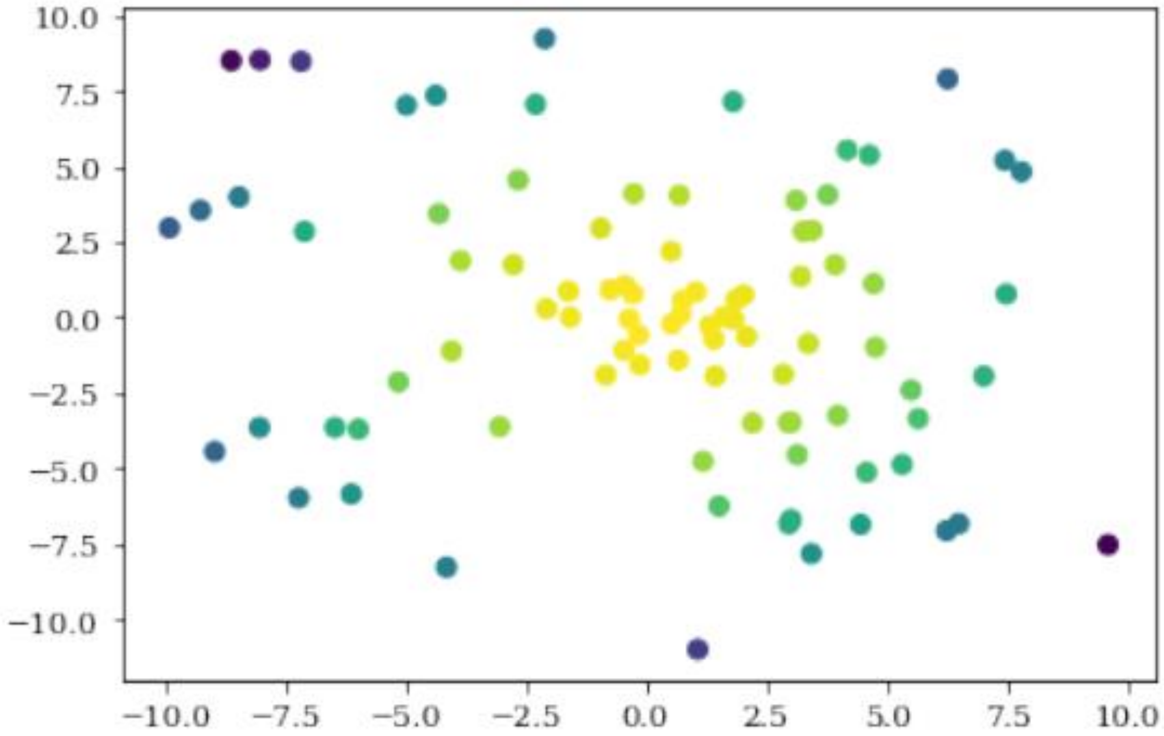


Extrinsic/Intrinsic parameter separation

[GitHub - oshaughn/RIFT_tutorials: RIFT tutorials]

RIFT example





Summary

- Enough independent samples following selected posterior distribution
- LALInference
  - Monte Carlo Markov Chain(MCMC)
  - Nested Sampling
- Bilby
  - Many sampling methods implemented Python library
- PyCBC Inference

PESummary

PESummary(post process)

- [PESummary | Home — PESummary 0.12.1+46.gfa604f0.dirty documentation (ligo.org)]()
- [PESummary: the code agnostic Parameter Estimation Summary page builder (arxiv.org)]()

```python
# import useful python packages
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import h5py
```

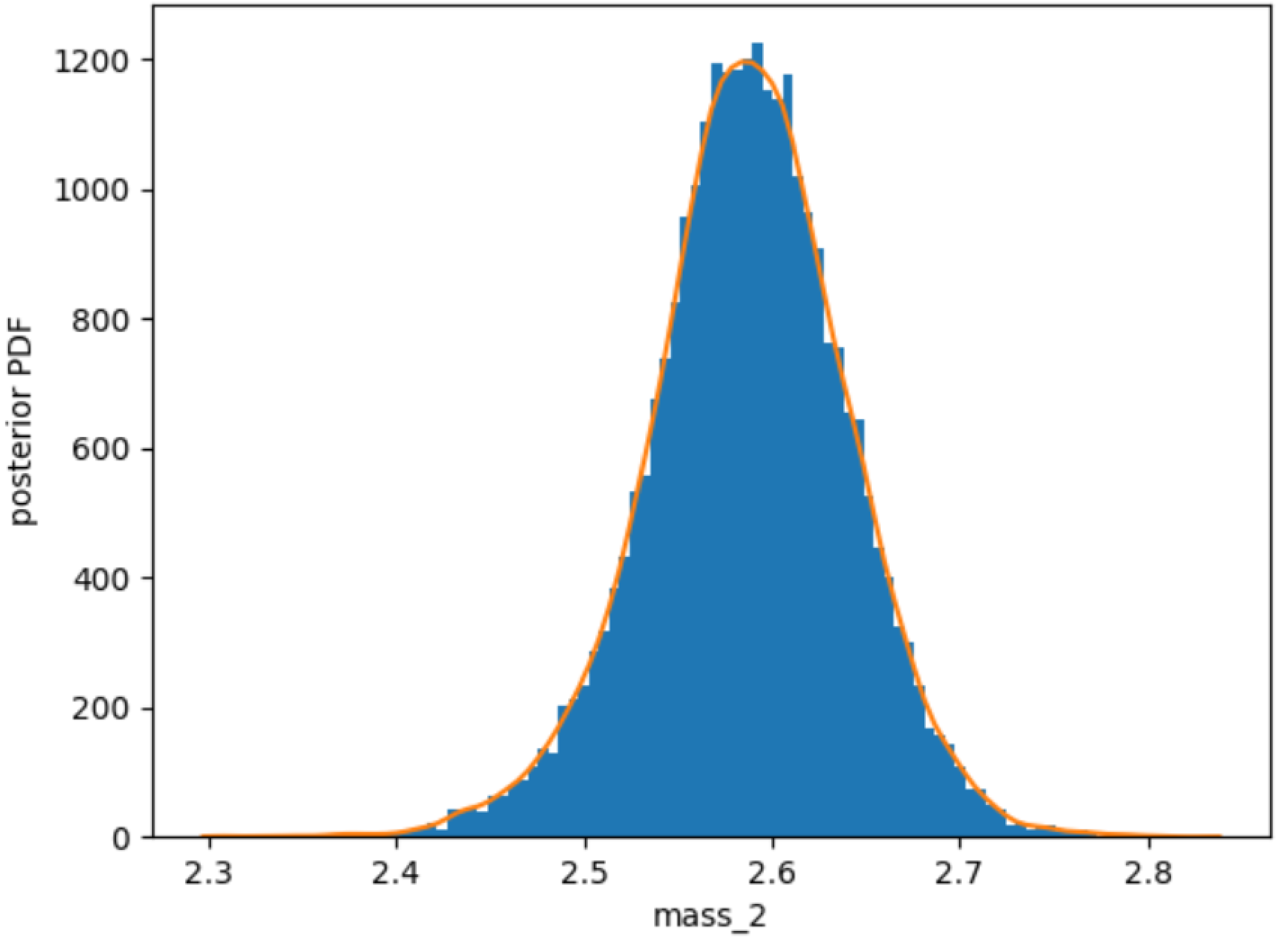
```python
# read in the data
fn = "GW190814_posterior_samples.h5"
data = h5py.File(fn,'r')

# print out parametrized waveform family names ("approximants" in LIGO jargon).
print('approximants:',data.keys())

# print out top-level data structures for one approximant. Here fore example we use the combined samples
# between IMRPhenomPv3HM and SEOBNRv4PHM. The data structure is the same for all approximants.
print('Top-level data structures:',data['combined'].keys())

# extract posterior samples for one of the approximants
posterior_samples = data['combined']['posterior_samples']
print('data structures in posterior_samples:',posterior_samples.dtype)
pnames = [item for item in posterior_samples.dtype.names]
print('parameter names:',pnames)

# get samples for one of the parameters
m2 = posterior_samples['mass_2_source']
print('mass_2 shape, mean, std =',m2.shape,m2.mean(),m2.std())

# smooth it
from scipy.stats.kde import gaussian_kde
hs = gaussian_kde(m2)

# histogram, and overlay the smoothed PDF
plt.figure()
h, b, o = plt.hist(m2,bins=100)
hsmoothed = hs(b)*len(m2)*(b[1]-b[0])
plt.plot(b,hsmoothed)
plt.xlabel('mass_2')
plt.ylabel('posterior PDF')
plt.show()

data.close()

# release memory for the data
#del data
```

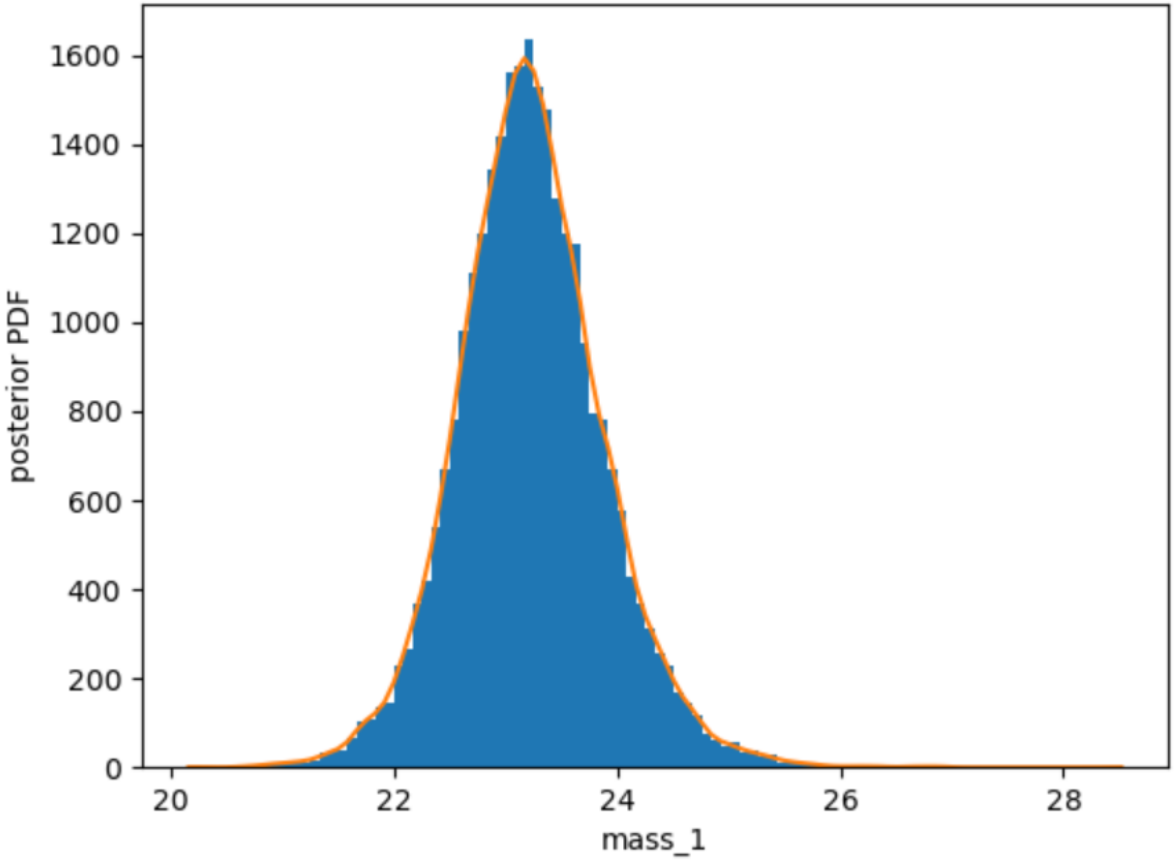
PESummary Example(GW190814)

PESummary | Home — PESummary 0.12.1+46.gfa604f0.dirty documentation (ligo.org)

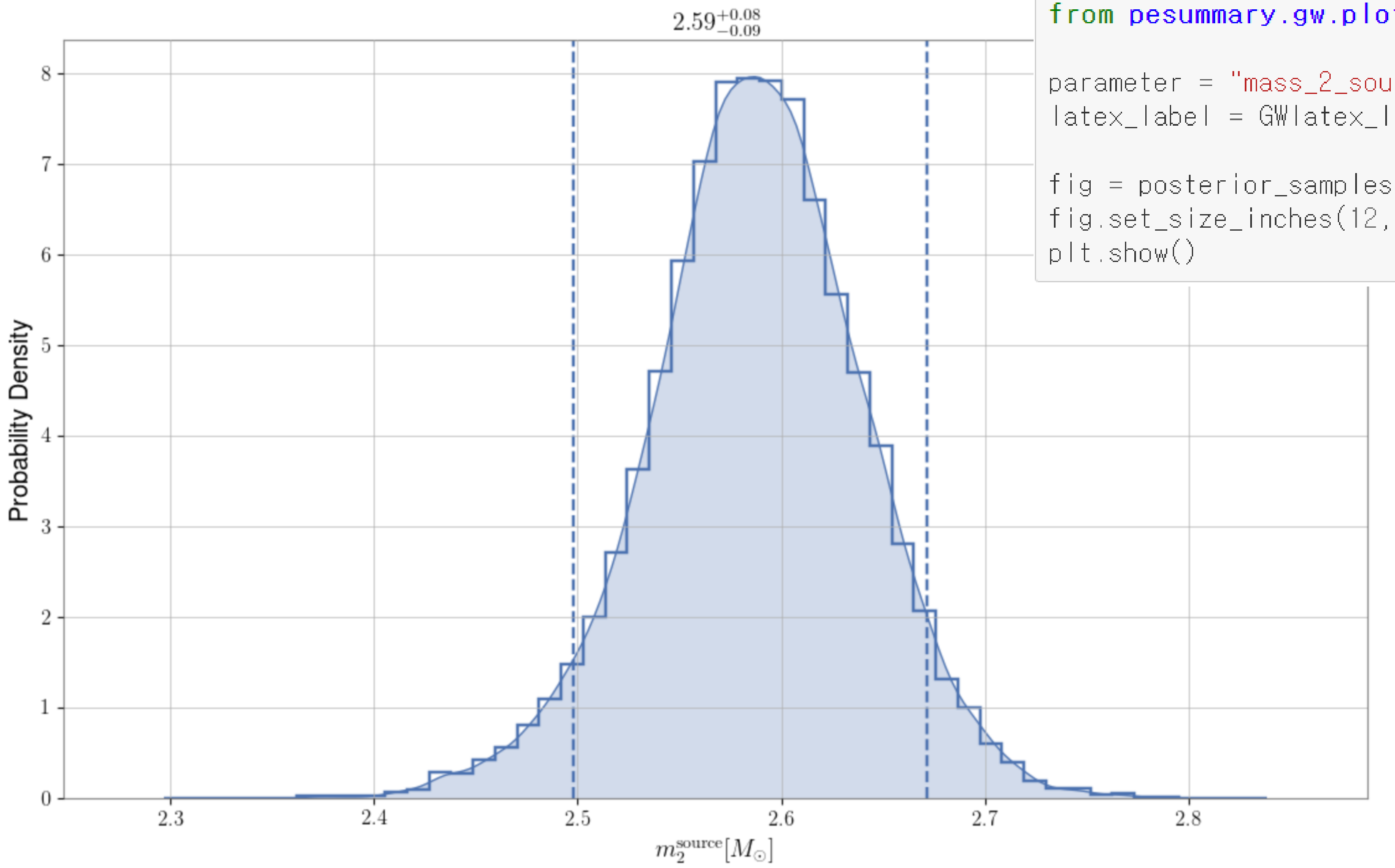
Run in anaconda

https://dcc.ligo.org/DocDB/0168/P2000183/010/GW190814_posterior_samples.html



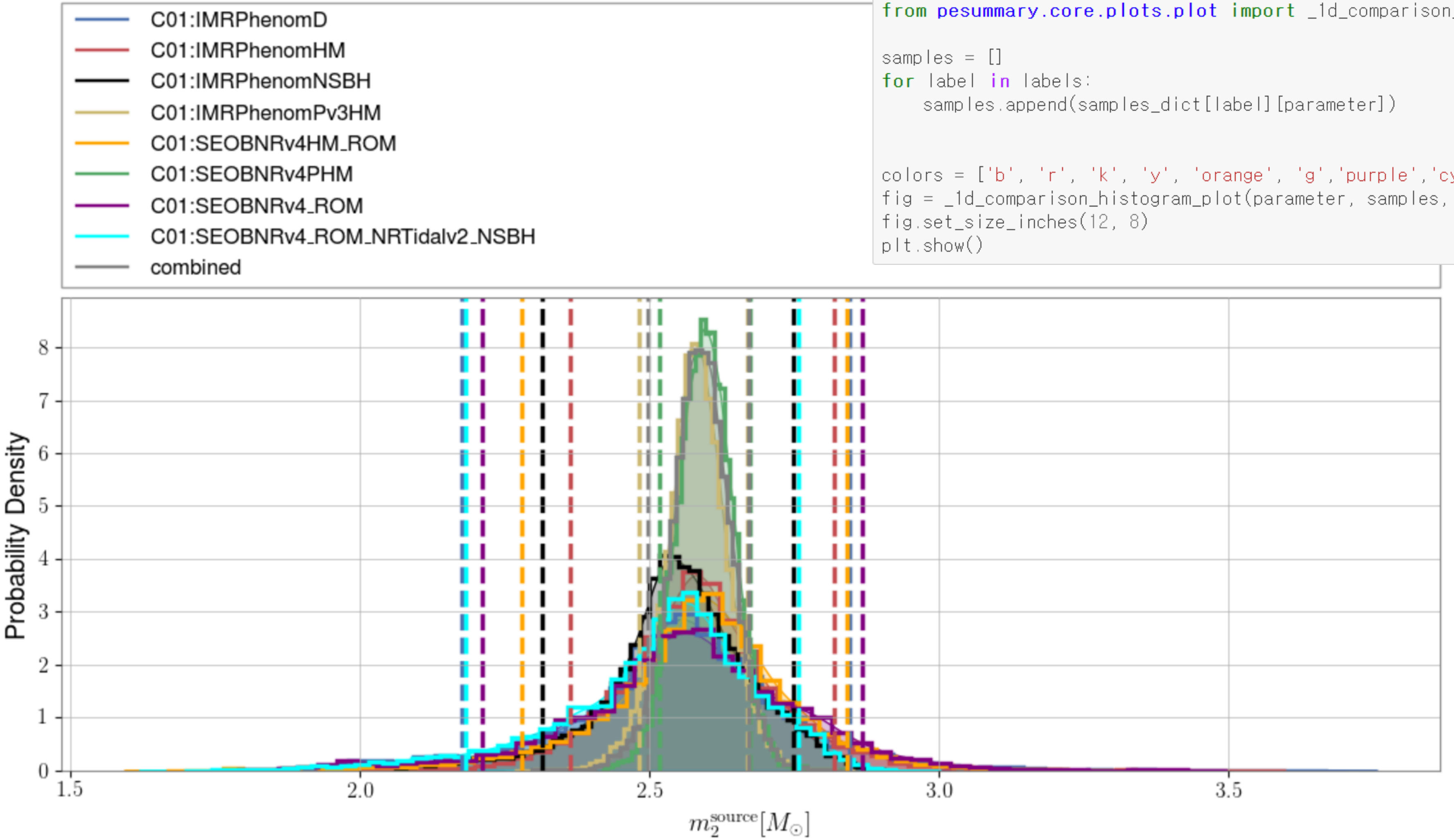


PESummary Example




```python
from pesummary.gw.plots.latex_labels import GWlatex_labels

parameter = "mass_2_source"
latex_label = GWlatex_labels[parameter]

fig = posterior_samples.plot(parameter, type="hist", kde=True)
fig.set_size_inches(12, 8)
plt.show()
```

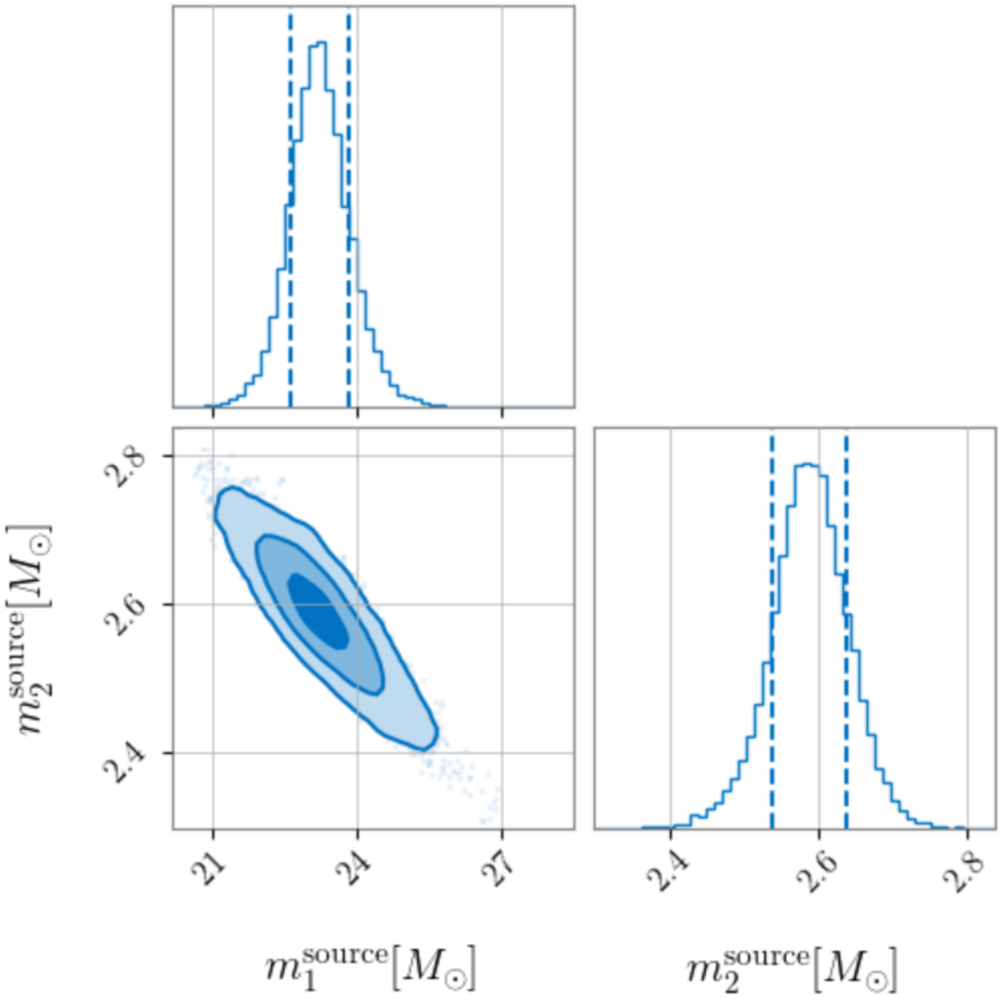
PESummary Example

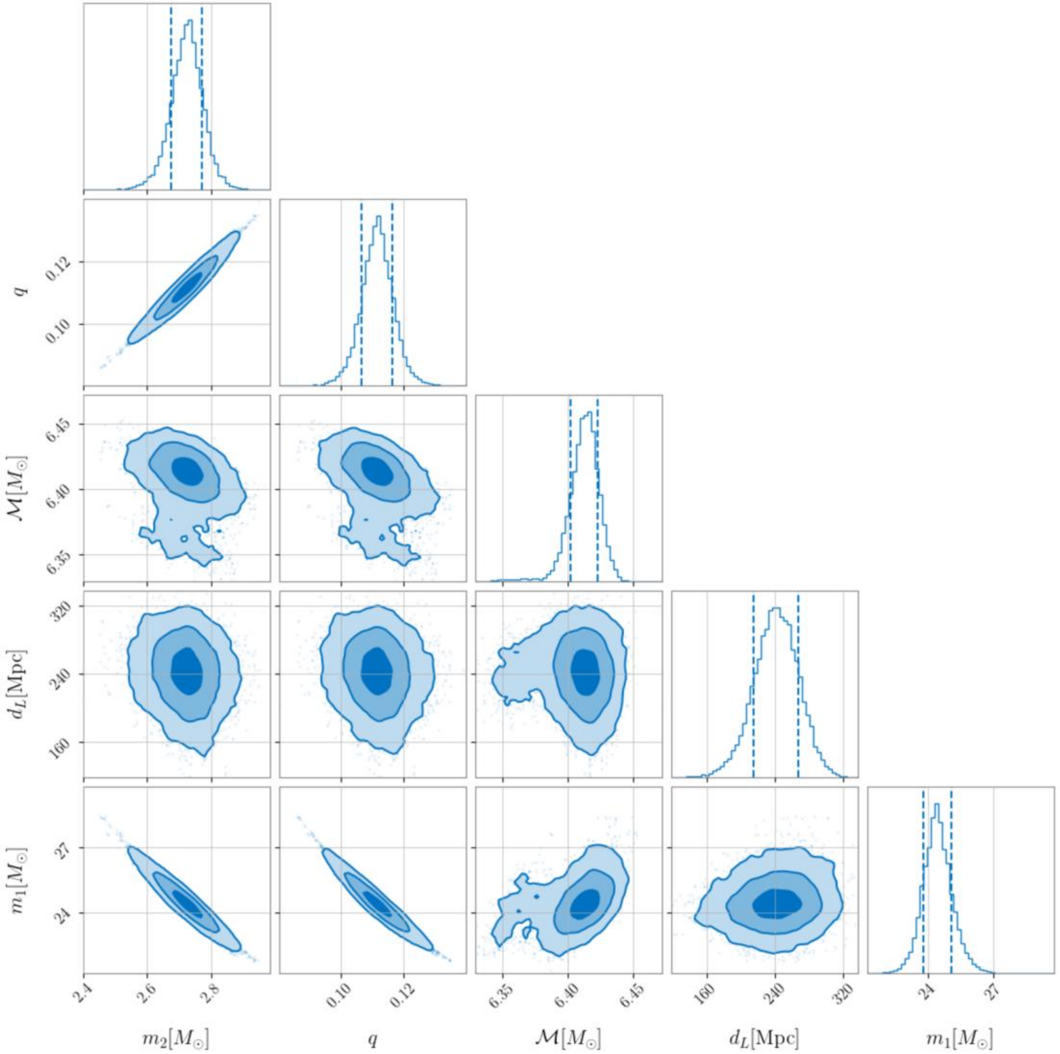



```
from pesummary.core.plots.plot import _1d_comparison_histogram_plot

samples = []
for label in labels:
    samples.append(samples_dict[label][parameter])


colors = ['b', 'r', 'k', 'y', 'orange', 'g','purple','cyan','grey','violet']
fig = _1d_comparison_histogram_plot(parameter, samples, colors, latex_label, sanitized_labels, kde=True)
fig.set_size_inches(12, 8)
plt.show()
```

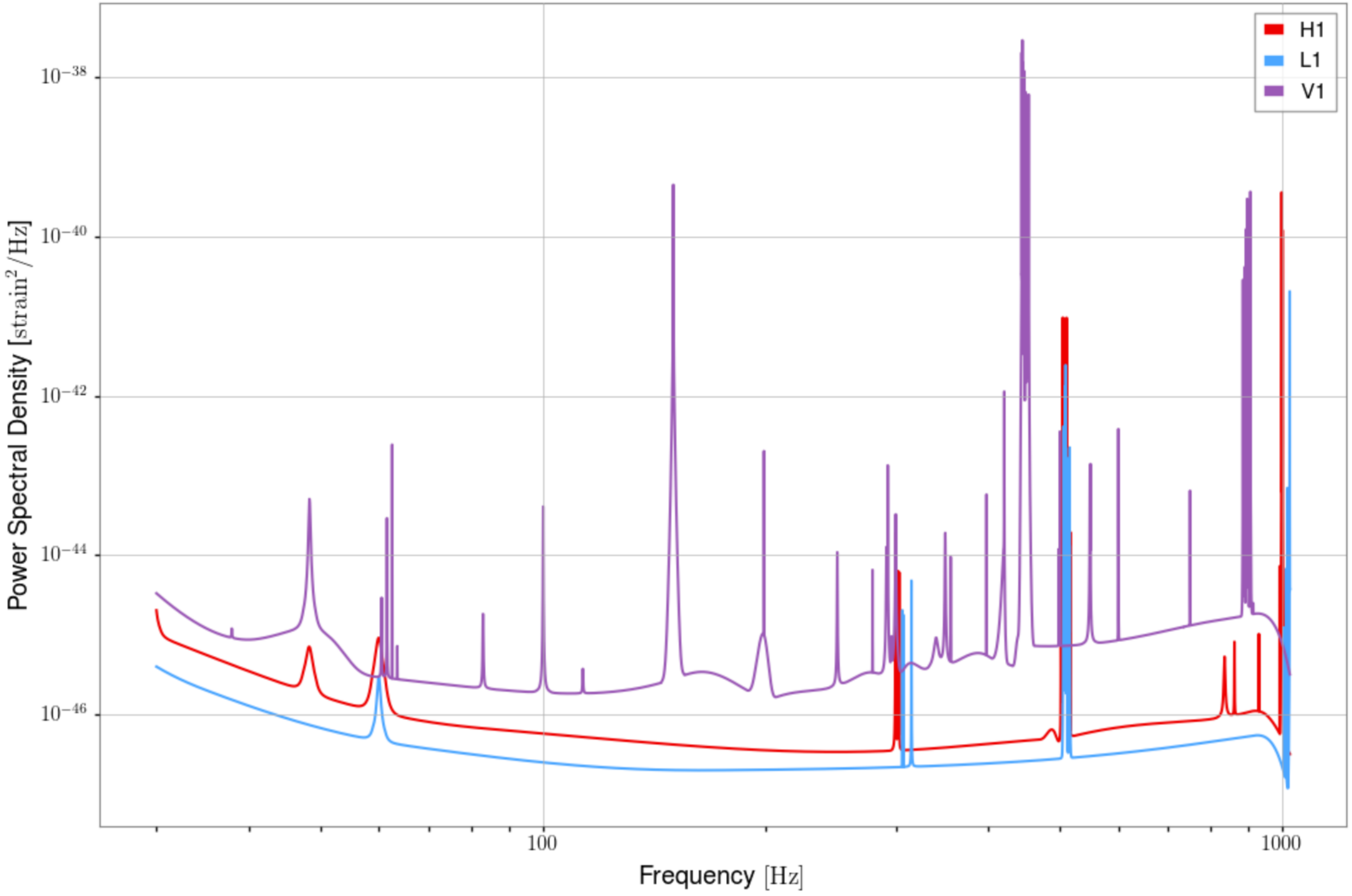
PESummary Example

```
fig = _make_corner_plot(posterior_samples, GWlatex_labels, corner_params)
plt.show()
corner_params : ['mass_1', 'mass_2', 'chirp_mass', 'mass_ratio', 'luminosity_distance']
```



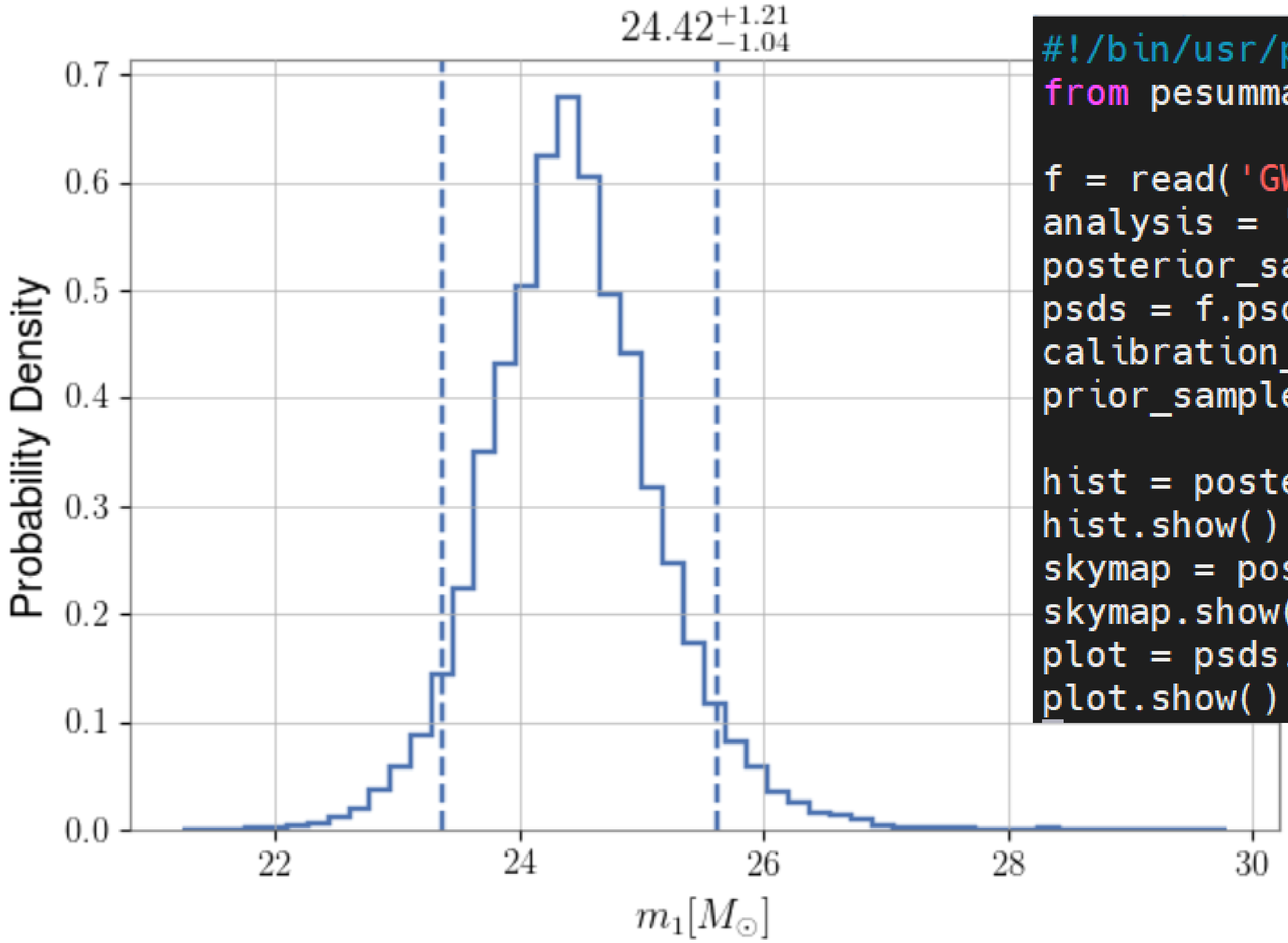


PESummary Example




```
psd = data.psd["combined"]
fig = psd.plot(fmin=30.0)
fig.set_size_inches(12, 8)
plt.show()
```

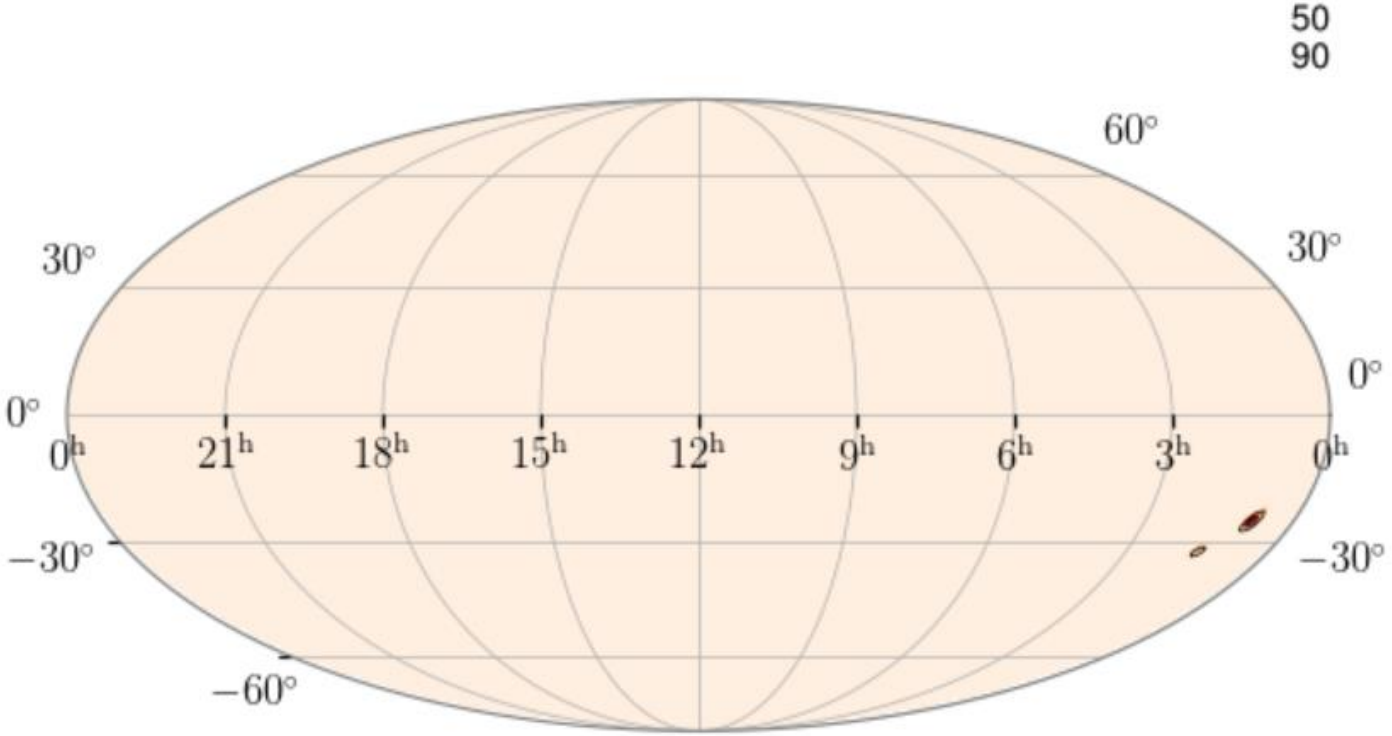
PESummary example



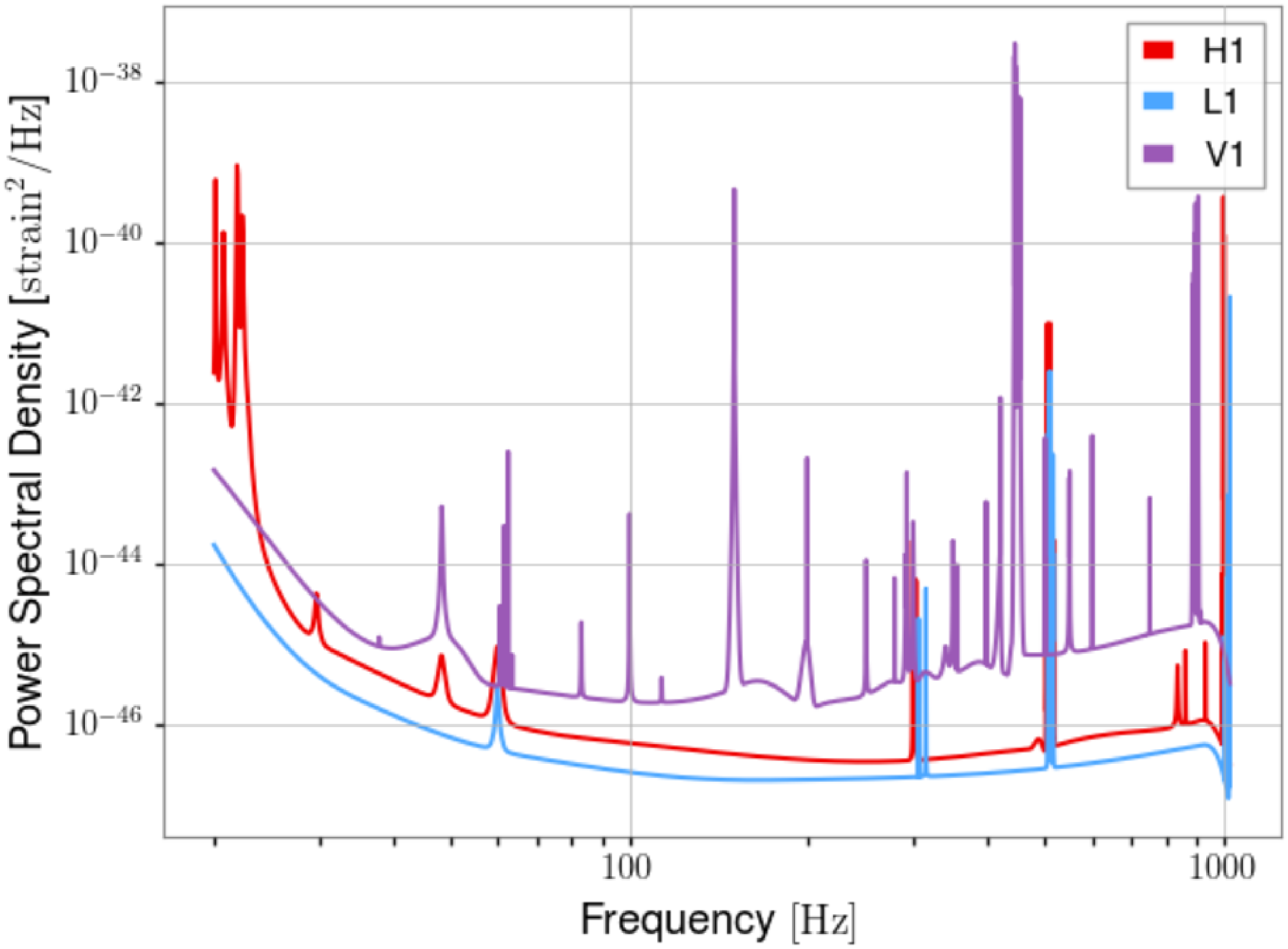

```python
#!/bin/usr/python
from pesummary.io import read

f = read('GW190814_posterior_samples.h5', package='gw')
analysis = 'combined'
posterior_samples = f.samples_dict[analysis]
psds = f.psd[analysis]
calibration_envelope = f.priors['calibration'][analysis]
prior_samples = f.priors['samples'][analysis]

hist = posterior_samples.plot('mass_1', type='hist')
hist.show()
skymap = posterior_samples.plot(type='skymap')
skymap.show()
plot = psds.plot(fmin=20)
plot.show()
```

PESummary example

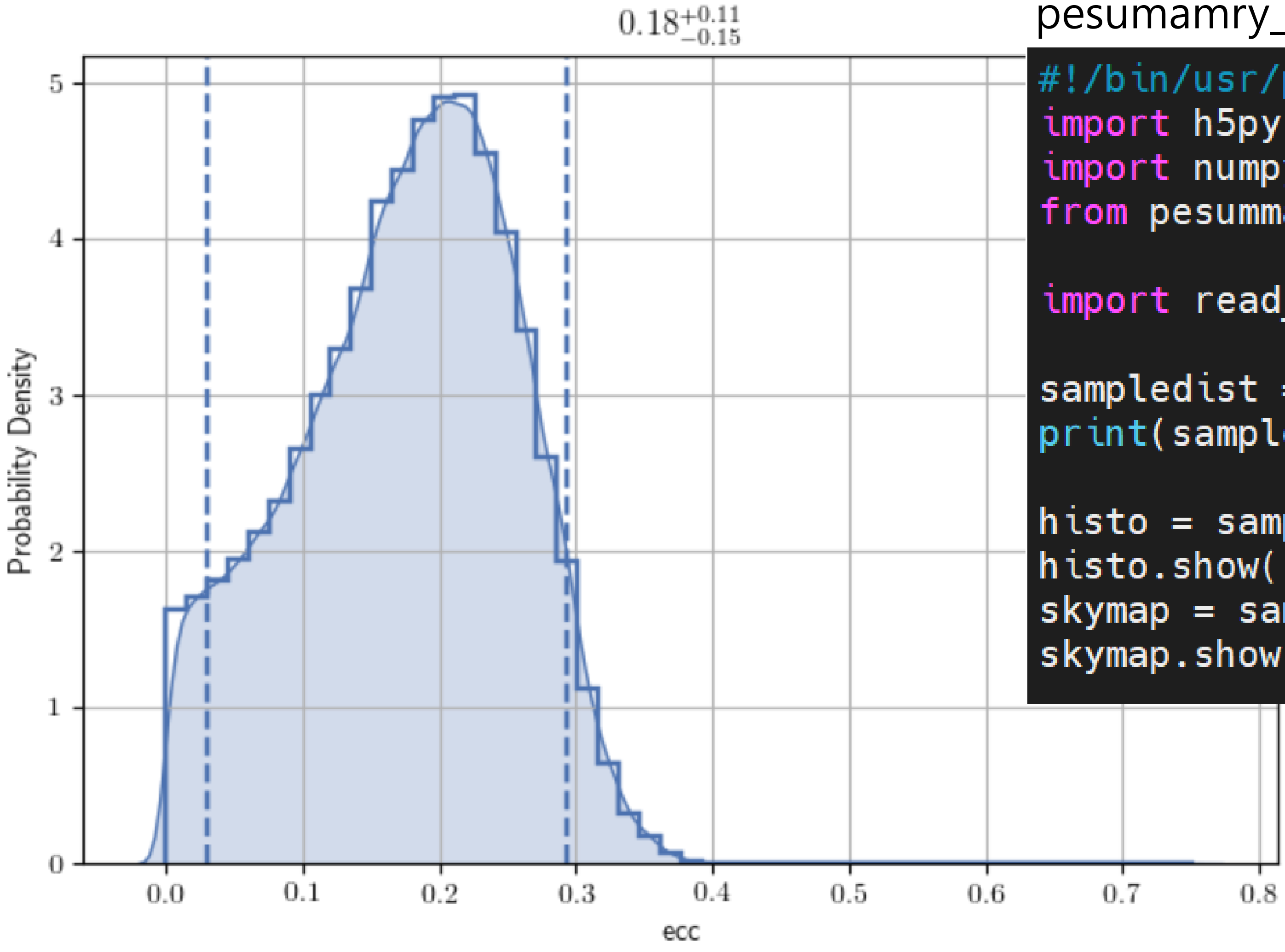
It takes long time, 300s/iteration, 200 iterations ~ 20hrs



PESummary Example



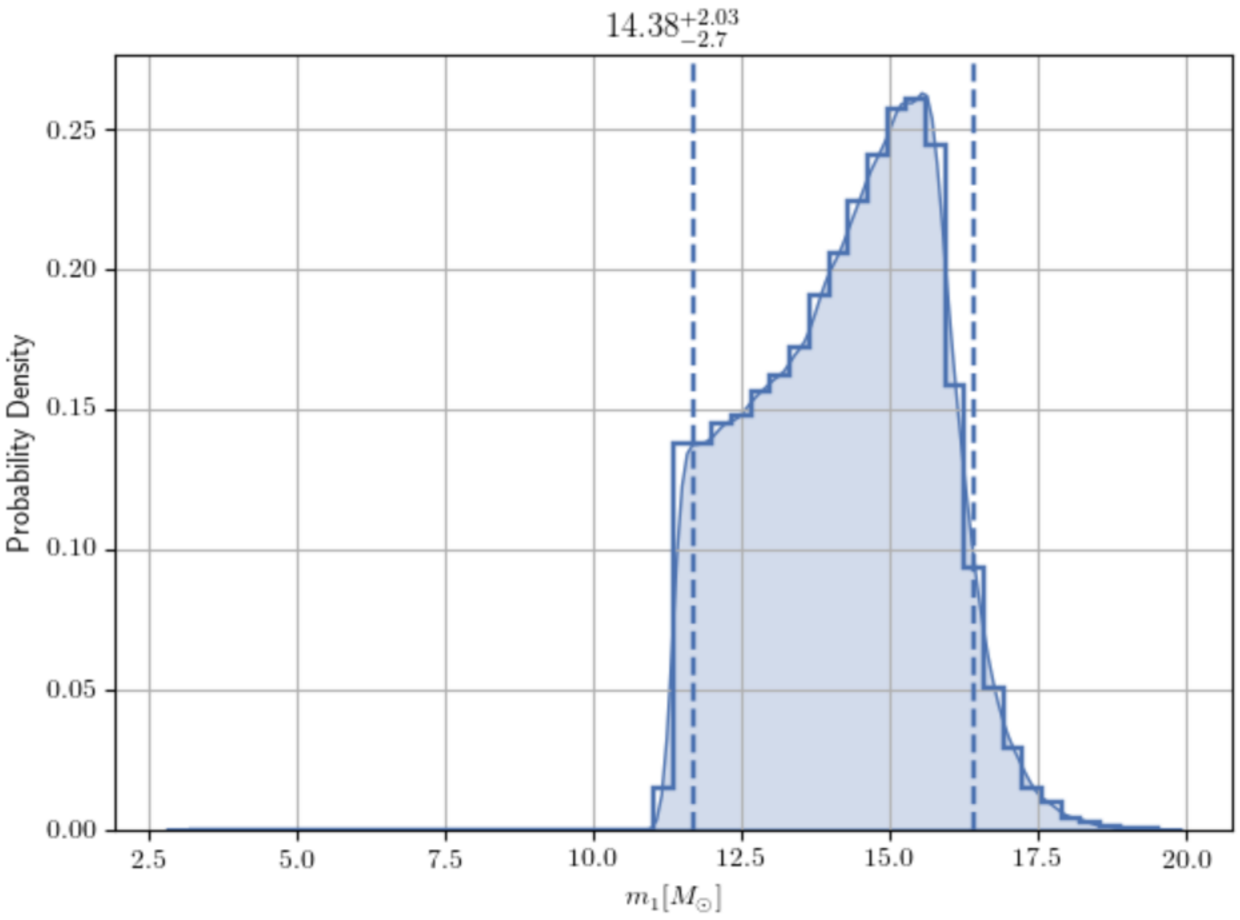
PESummary Example(legacy)

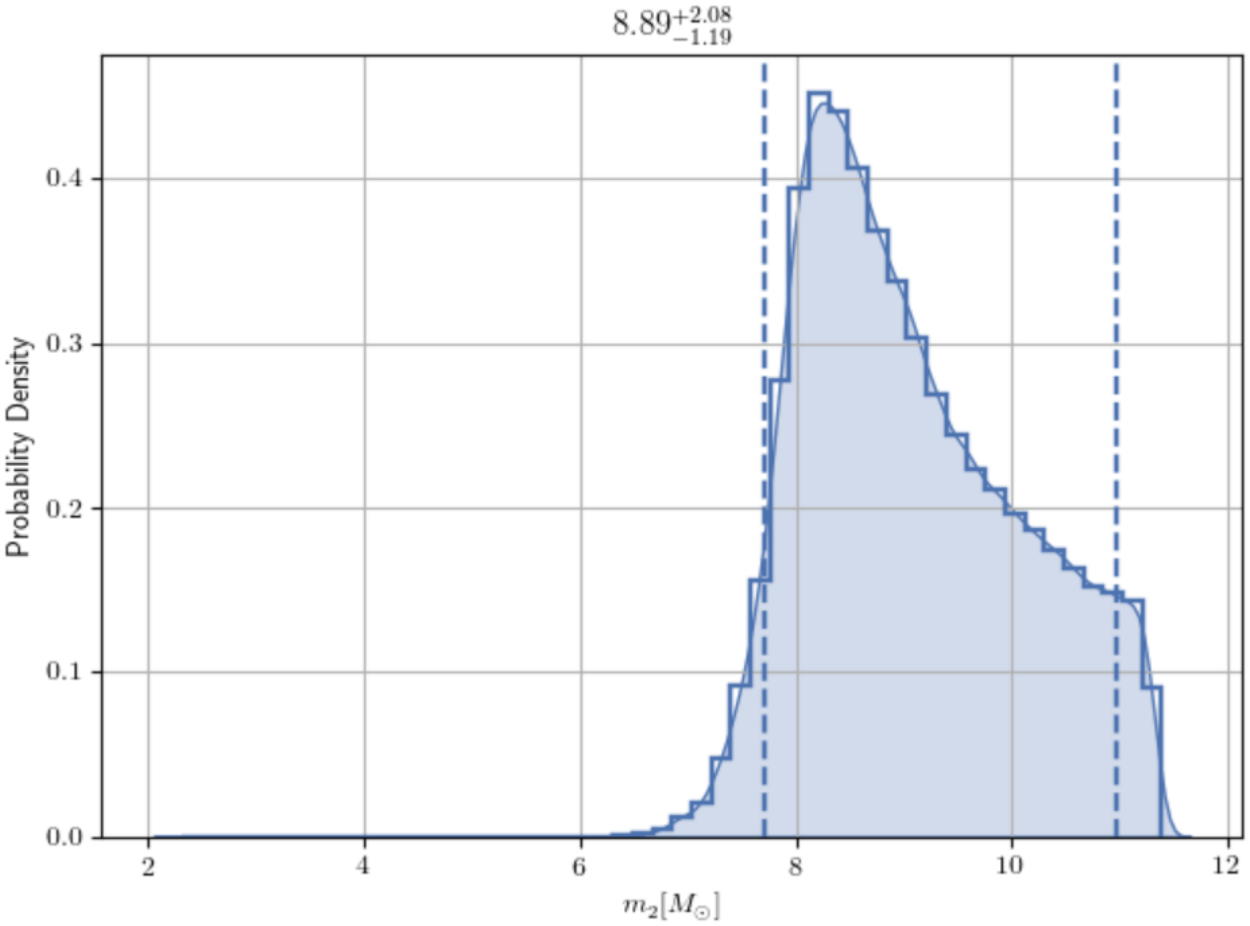


pesumamry_legacy.py

```
#!/bin/usr/python
import h5py
import numpy as np
from pesummary.utils.samples_dict import SamplesDict

import read_legacy as legacy

sampledist = legacy.read_h5('PTMCMC.output.20193-0.h5')
print(samplesdict)

histo = samplesdict.plot('ecc', type='hist', kde=True)
histo.show()
skymap = samplesdict.plot(type='skymap')
skymap.show()
```

PESummary Example(legacy)





Fisher Information Matrix

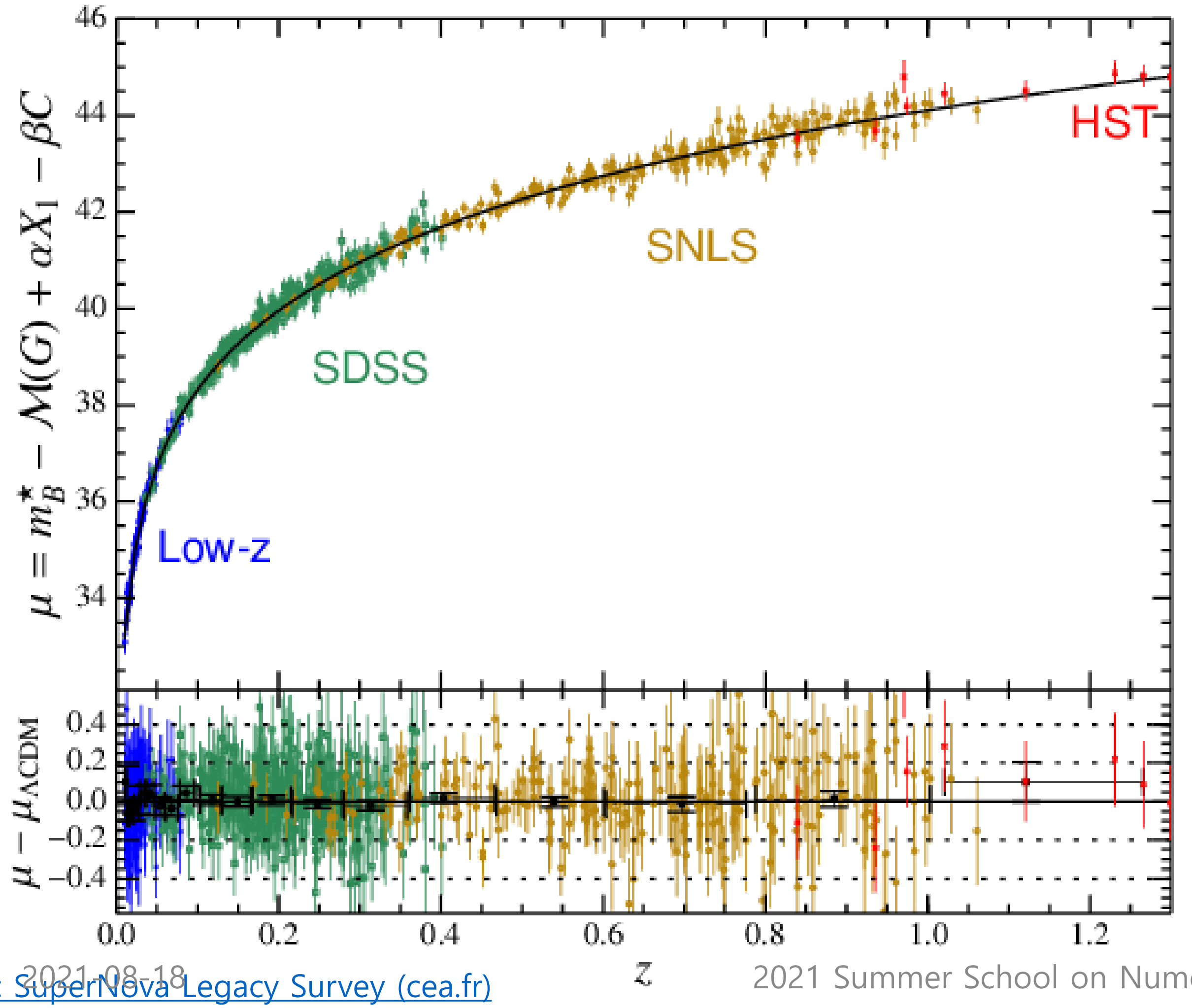
Fisher information matrix

- Likelihood $p(M|D) \propto e^{-\frac{1}{2}L_p}$
- log-Likelihood $\ln p(M|D) \propto -\frac{1}{2}L_p\left(\vec{\theta}, D\right)$
- Taylor expansion around peak parameter $\hat{\theta}$
- $\ln p(M|D) \propto -\frac{1}{2}L_p\left(\vec{\theta}, D\right) = -\frac{1}{2}L_p(\hat{\theta}, D) - \frac{1}{2}\frac{\partial L_P\left(\vec{\theta},D\right)}{\partial \theta_j}(\theta_j - \hat{\theta}_j) - \frac{1}{2}\frac{1}{2}\frac{\partial^2 L_P\left(\vec{\theta},D\right)}{\partial \theta_j \partial \theta_k}(\theta_j - \hat{\theta}_j)(\theta_k - \hat{\theta}_k) + \cdots = -\frac{1}{2}L_p(\hat{\theta}, D) - \frac{1}{2}\frac{1}{2}\frac{\partial^2 L_P\left(\vec{\theta},D\right)}{\partial \theta_j \partial \theta_k}(\theta_j - \hat{\theta}_j)(\theta_k - \hat{\theta}_k) + \cdots$

Likelihood for Data Fitting

- Measured values y_{i_b} with Gaussian Error Distribution
- $\chi^2 = \sum_{i=1}^{B} \sum_{i_b} \frac{\left(f_b(\theta) - y_{i_b}\right)^2}{\sigma_b^2}$
- Observation probability of data y_{i_b} if parameter is θ
- $P\left(y_{i_b}|\theta\right) \propto e^{-\frac{1}{2}\chi^2}$: Likelihood
- If one assume uniform prior for parameters, Bayes' theorem says
- $P\left(\theta|y_{i_b}\right) = P\left(y_{i_b}|\theta\right)$

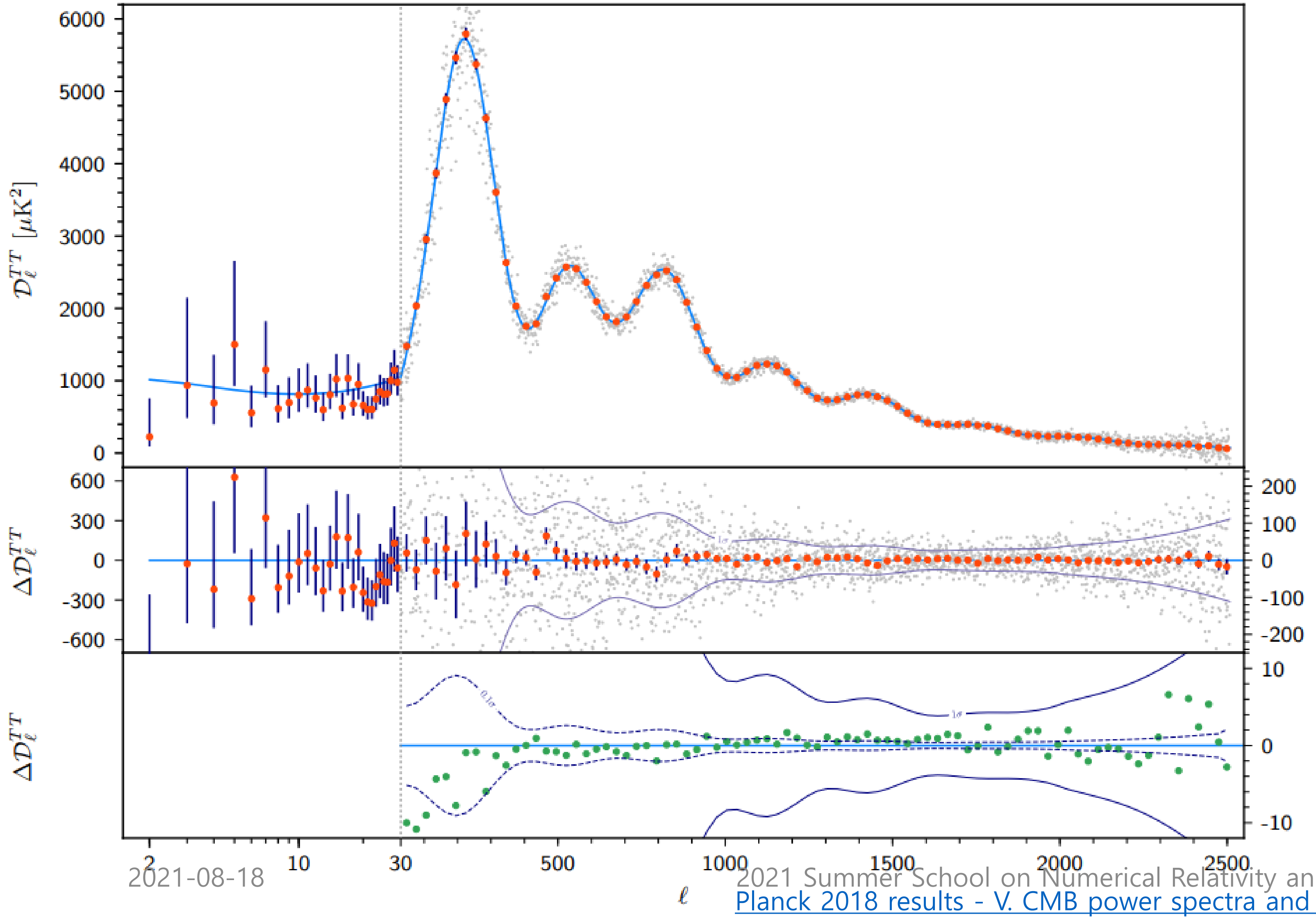
Super Novae Data



- Number of bins ~ number of observed red-shift values
- $f_b(p)$ ~ fitting function for each observation bins
- y_{i_b} ~ observed data set for each observation bins : distance

[SNLS : SuperNova Legacy Survey (cea.fr)]

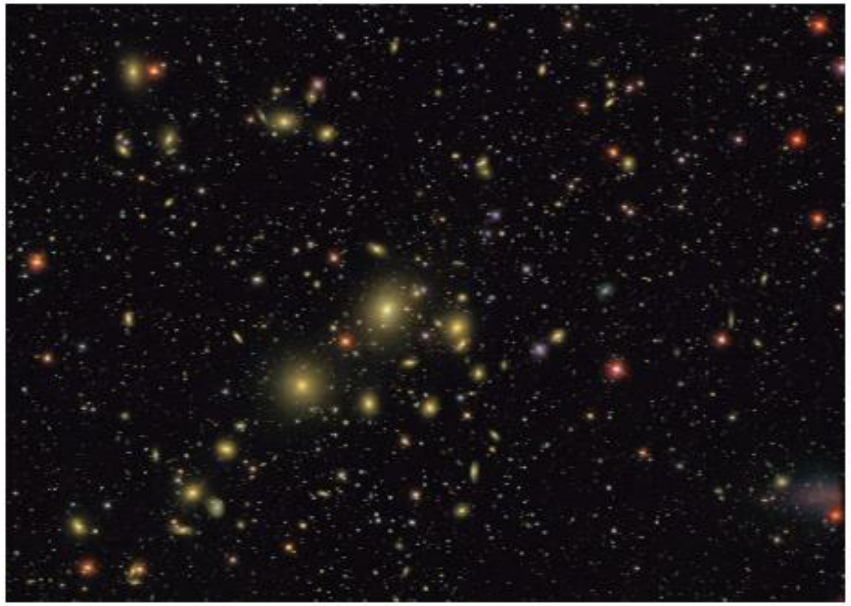
Baryon Acoustic Oscillation

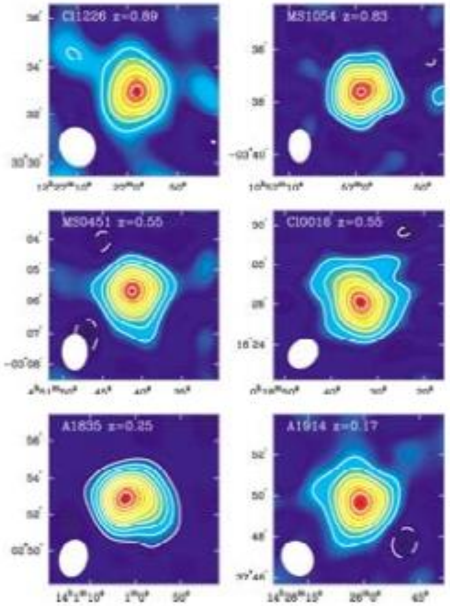


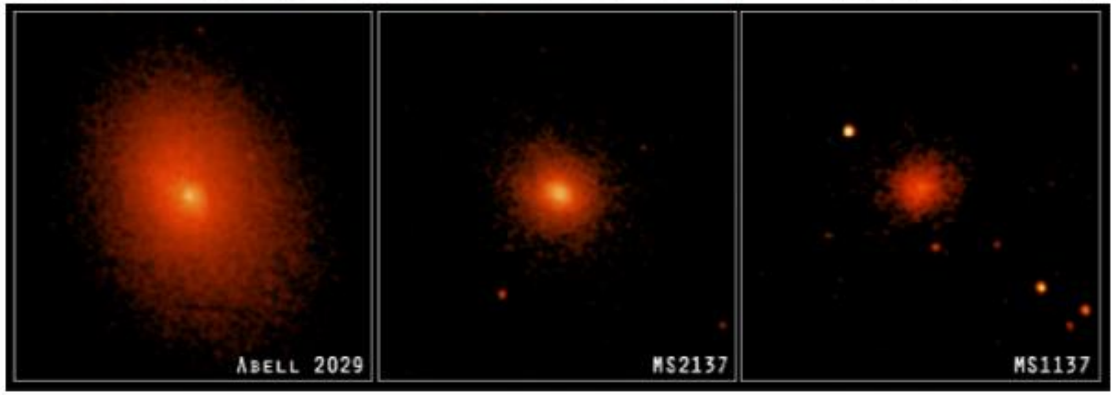
- Number of bins ~ number of observed angular frequency
- $f_b(p)$ ~ fitting function for each observation bins
- y_{i_b} ~ observed data set for each observation bins : amplitude

[Planck 2018 results - V. CMB power spectra and likelihoods | Astronomy & Astrophysics (A&A) (aanda.org)](aanda.org)

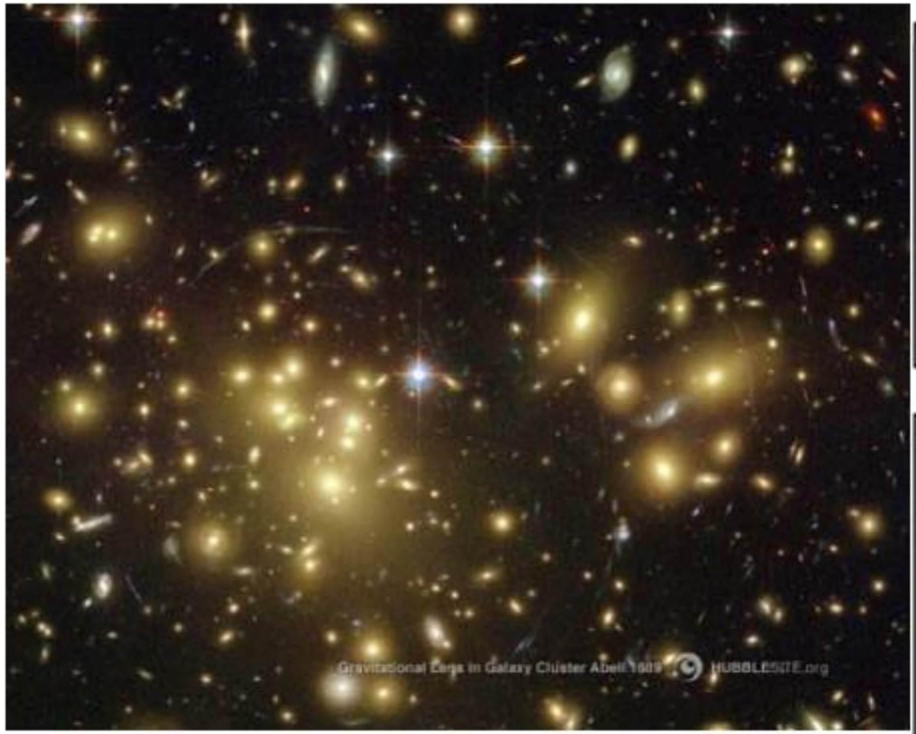
Galaxy Cluster Counter

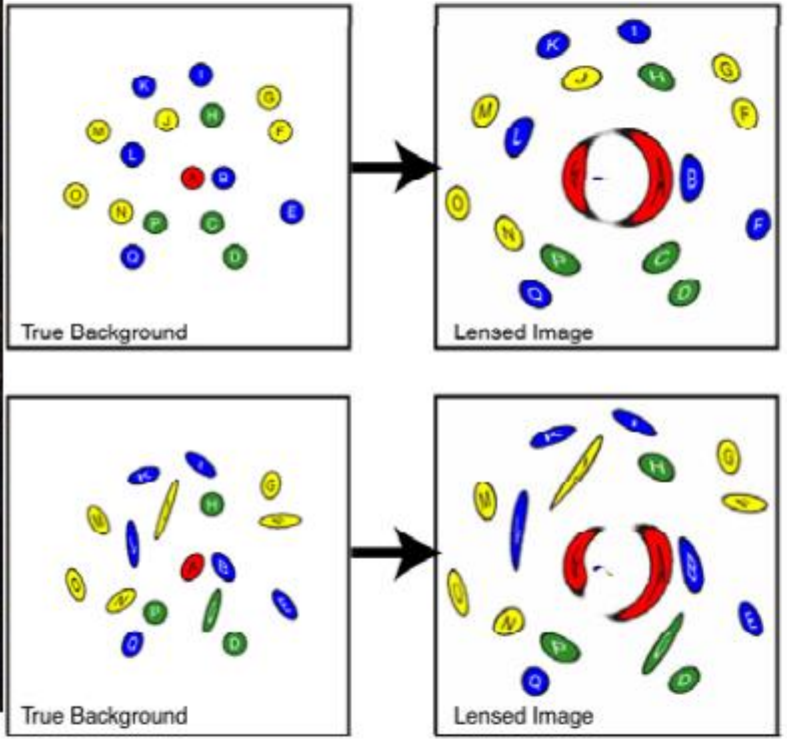






Weak Lensing





[Microsoft Word - DETF_Final.doc (arxiv.org)](#)

Fisher Matrix

- $\langle \chi^2(\theta) \rangle = \langle \chi^2 \rangle + \left\langle \frac{\partial \chi^2}{\partial \theta_j} \right\rangle \delta\theta_j + \frac{1}{2} \left\langle \frac{\partial^2 \chi^2}{\partial \theta_j \partial \theta_k} \right\rangle \delta\theta_j \delta\theta_k + \cdots$
- $\langle \cdot \rangle$ means evaluate at extreme parameter θ_0
- $\left\langle \frac{\partial \chi^2}{\partial \theta_j} \right\rangle_{\theta_0} = 0$
- $\langle \chi^2(\theta) \rangle = \langle \chi^2 \rangle + \frac{1}{2} \left\langle \frac{\partial^2 \chi^2}{\partial \theta_j \partial \theta_k} \right\rangle \delta\theta_j \delta\theta_k + \cdots = \langle \chi^2 \rangle + F_{jk} \delta\theta_j \delta\theta_k + \cdots$
- $F_{jk} = \frac{1}{2} \left\langle \frac{\partial^2 \chi^2}{\partial \theta_j \partial \theta_k} \right\rangle = \sum_b \frac{N_b}{\sigma_b^2} \frac{\partial f_b}{\partial \theta_j} \frac{\partial f_b}{\partial \theta_k}$ proportional to gradient of measure function

Example Simulation

- We want to fit a data with $y = a\theta$
- Sample Data is generated as follows
- $y_i = a\theta_i + N(a\theta_i, \sigma_i)$
- Each data point y_i is average of N_i random variable $r = a\theta_i + N(a\theta_i, \sigma_i)$
- σ_i vary in range (s_0, s_1)

Example Simulation

```python
#Sample calculation for Fisher Matrix meaning
# sample data is generated by y = a*theta + n
#  a = 1, 0<theta<1, n ~ N(a*theta, sigma)
#  0.1 < sigma < 0.3
# for each theta, averaged value is observed
import os
import numpy as np
import scipy
import matplotlib.pyplot as plt
```

```python
randFloat = np.random.rand(5)
#print("randfloat between 0.0 and 1.0:", randFloat)
def chi2(x, y, s, a, N):
    ch0 = 0
    for i in range(N):
        ch0 += ((a*x[i] - y[i])/s[i])**2
    return ch0
```

```python
def fisher(a, s0, s1):
    # equal spaced theta points
    N = 100
    nb = 10
    dth = 1.0/N
    theta = []
    for i in range(N):
        theta.append(i*dth)
    #print("theta : ", theta)
    # random sigma values between s0 and s1
    #s0 = 0.1
    #s1 = 0.3
    sigmas = s0 + np.random.rand(N)*(s1 - s0)
    #print("sigma : ", sigmas)
    # observation data as y = a * theta + noise
    data = []
    y = []
    #a = 1.0
    for i in range(N):
        y0 = a*theta[i]
        yb = []
        for j in range(nb):
            r = np.random.normal(y0,sigmas[i])
            yb.append(r)
        data.append(np.mean(yb))
    #print("average = ", np.mean(data), ", stddev = ", np.std(data))
    #print("data :", data)
```

```python
    a0, b0 = np.polyfit(theta, data, 1)
    for i in range(N):
        y.append(a0*theta[i])
    print("best fit a = ", a0, ", b = ", b0)
    scales = np.linspace(-5.0, 5.0, N)
    #print("scales : ", scales)
    chi2s = []
    for i in range(N):
        chi2s.append(chi2(theta, data, sigmas, scales[i], N))
    #print("chi2 : ", chi2s)
    plt.scatter(theta, data)
    plt.xlabel("theta")
    plt.ylabel("y")
    plt.title("y = theta + n")
    plt.show()

    plt.plot(theta, data)
    plt.xlabel("theta")
    plt.ylabel("y")
    plt.title("y = theta + n")
    plt.show()

    #plt.plot(theta, data)
    plt.xlabel("theta")
    plt.ylabel("y")
    plt.title("y = theta + n")
    plt.scatter(theta, data, color="black")
    plt.plot(theta, y, label="Linear Fit", color="red")
    plt.legend()
    plt.show()

    plt.plot(scales, chi2s)
    plt.xlabel("slope")
    plt.title("chi-square")
    plt.show()
    return chi2s, scales
```

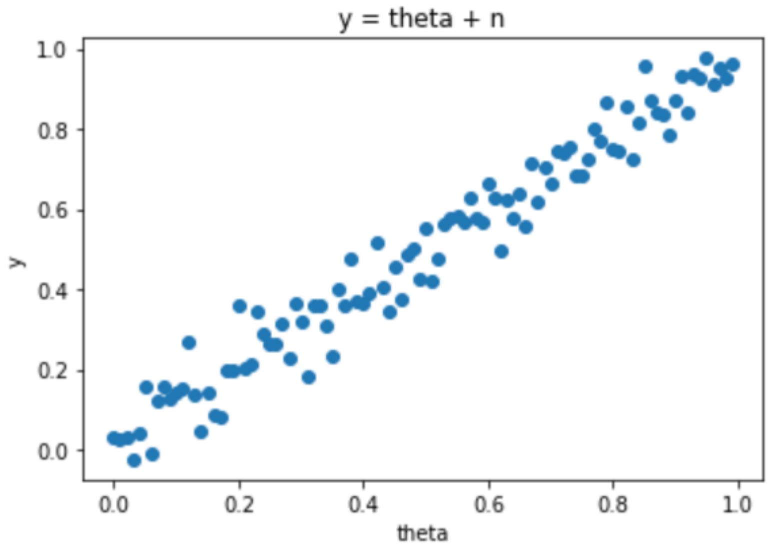
```python
chi2s1, scales1 = fisher(1.0, 0.1, 0.3)
```

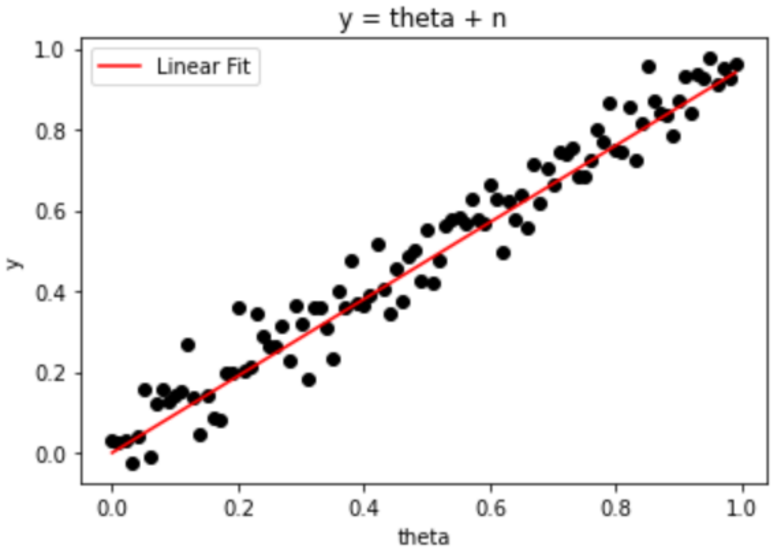
```python
chi2s2, scales2 = fisher(1.0, 0.1, 0.5)
```

```python
chi2s3, scales3 = fisher(1.0, 0.5, 0.8)
```

```python
chi2s4, scales4 = fisher(1.0, 1.0, 3.0)
```

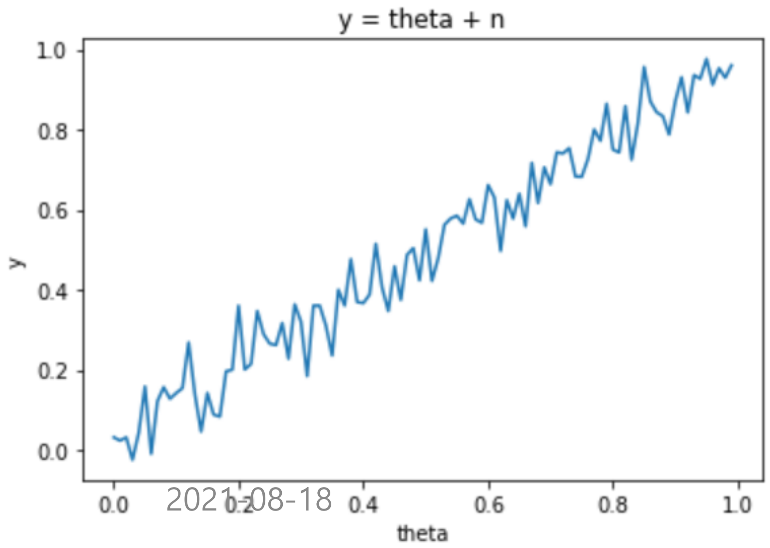
$$a = 1, s_0 = 0.1, s_1 = 0.3$$

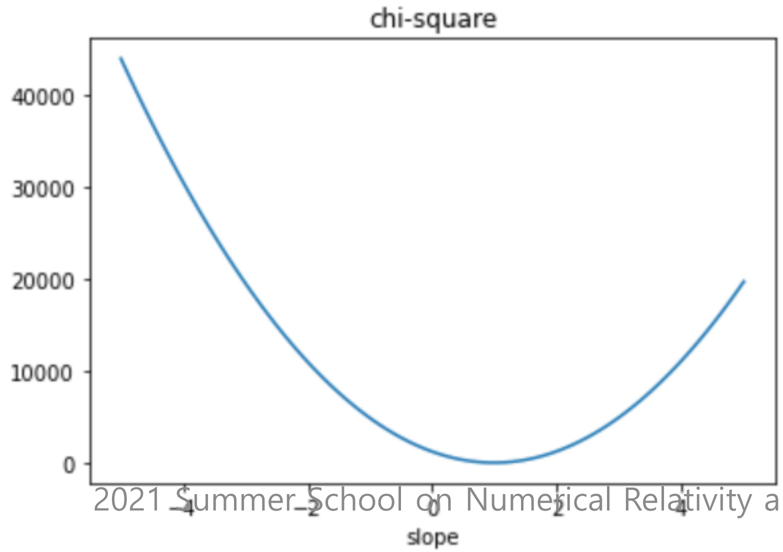




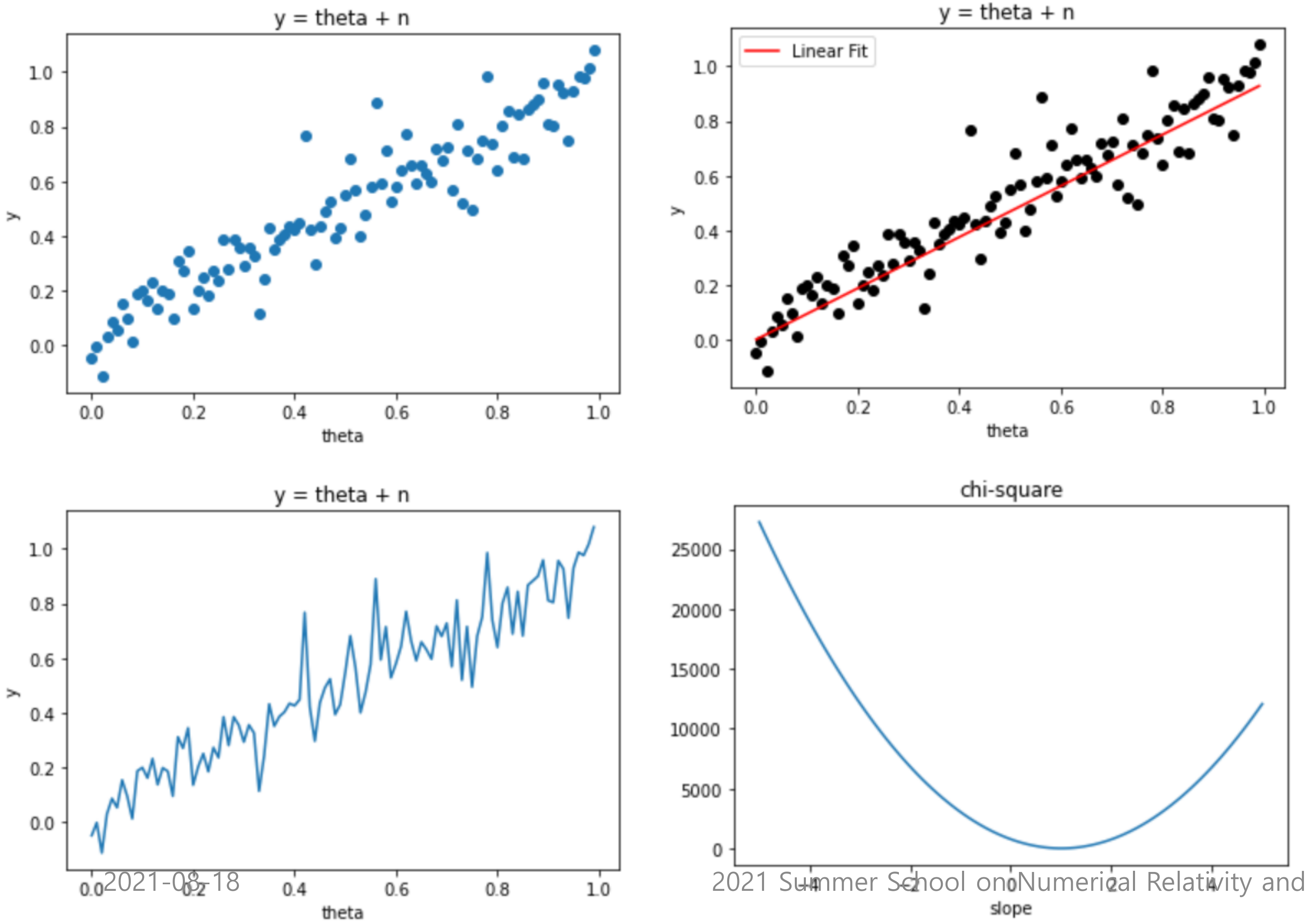

```
chi2s1, scales1 = fisher(1.0, 0.1, 0.3)
```

```
best fit a =  0.9507639434837137 , b =  0.021531902251367096
```





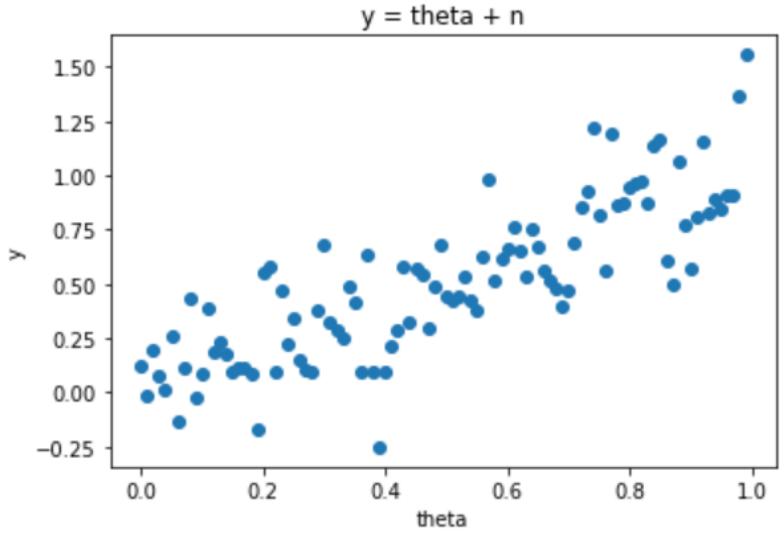
$a = 1, s_0 = 0.1, s_1 = 0.5$

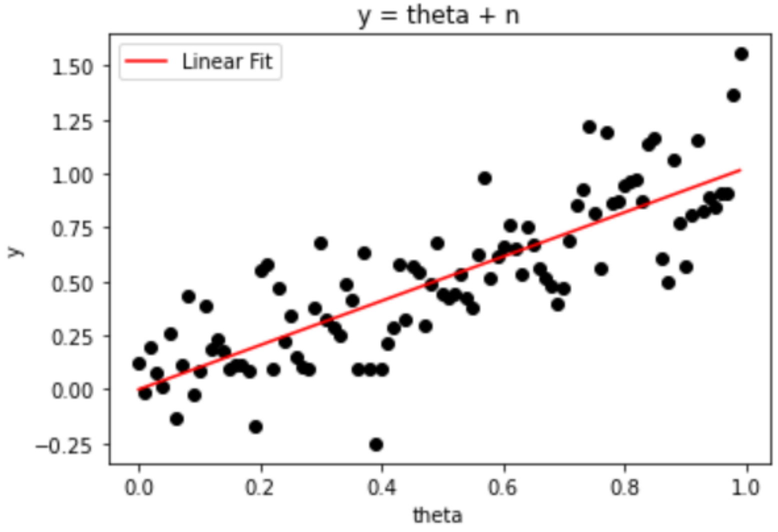



```
chi2s2, scales2 = fisher(1.0, 0.1, 0.5)
```

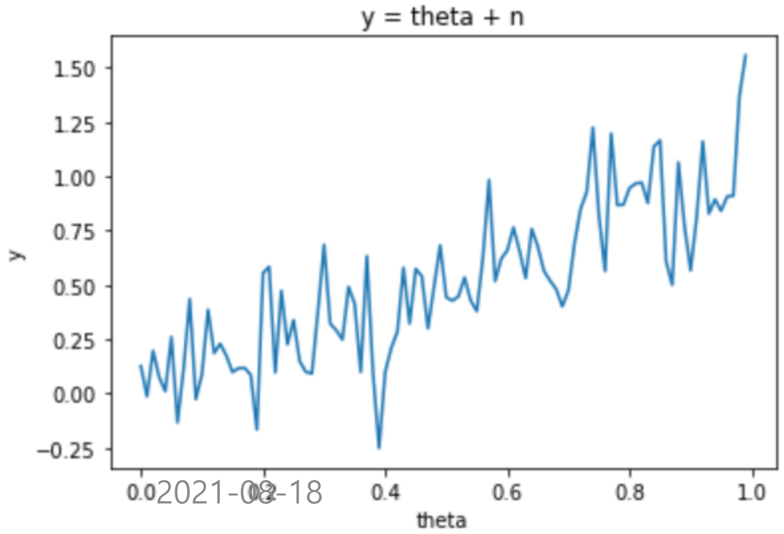
```
best fit a =  0.9376330047628946 , b =  0.03614953147962345
```

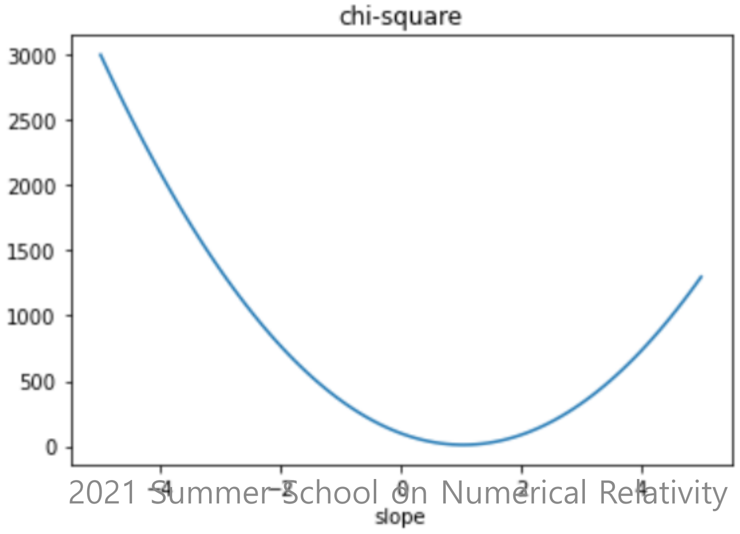
$$a = 1, s_0 = 0.5, s_1 = 0.8$$



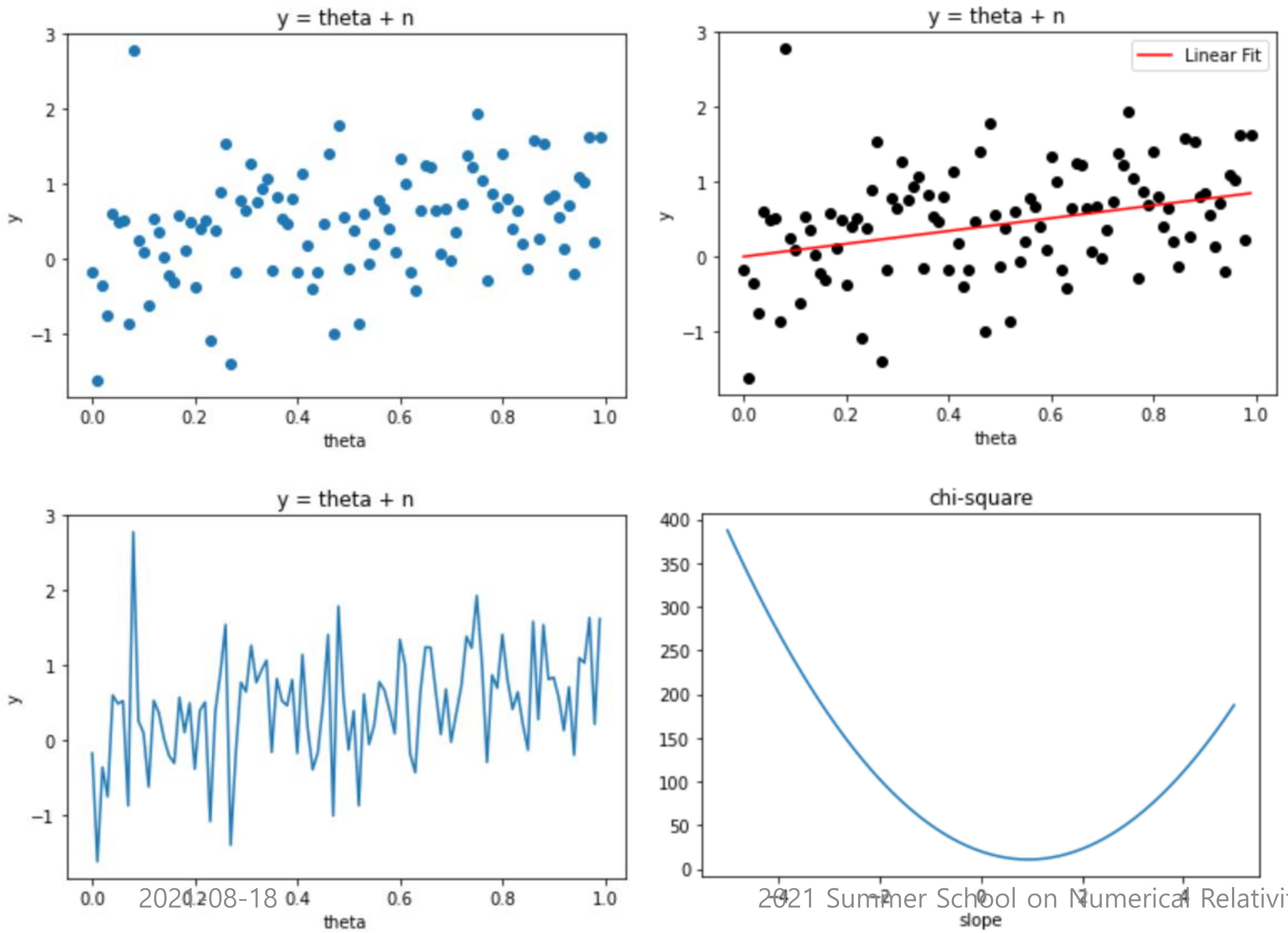



```
chi2s3, scales3 = fisher(1.0, 0.5, 0.8)

best fit a =  1.0247573110158383 , b =  0.005213721534364346
```





$a = 1, s_0 = 1.0, s_1 = 3.0$

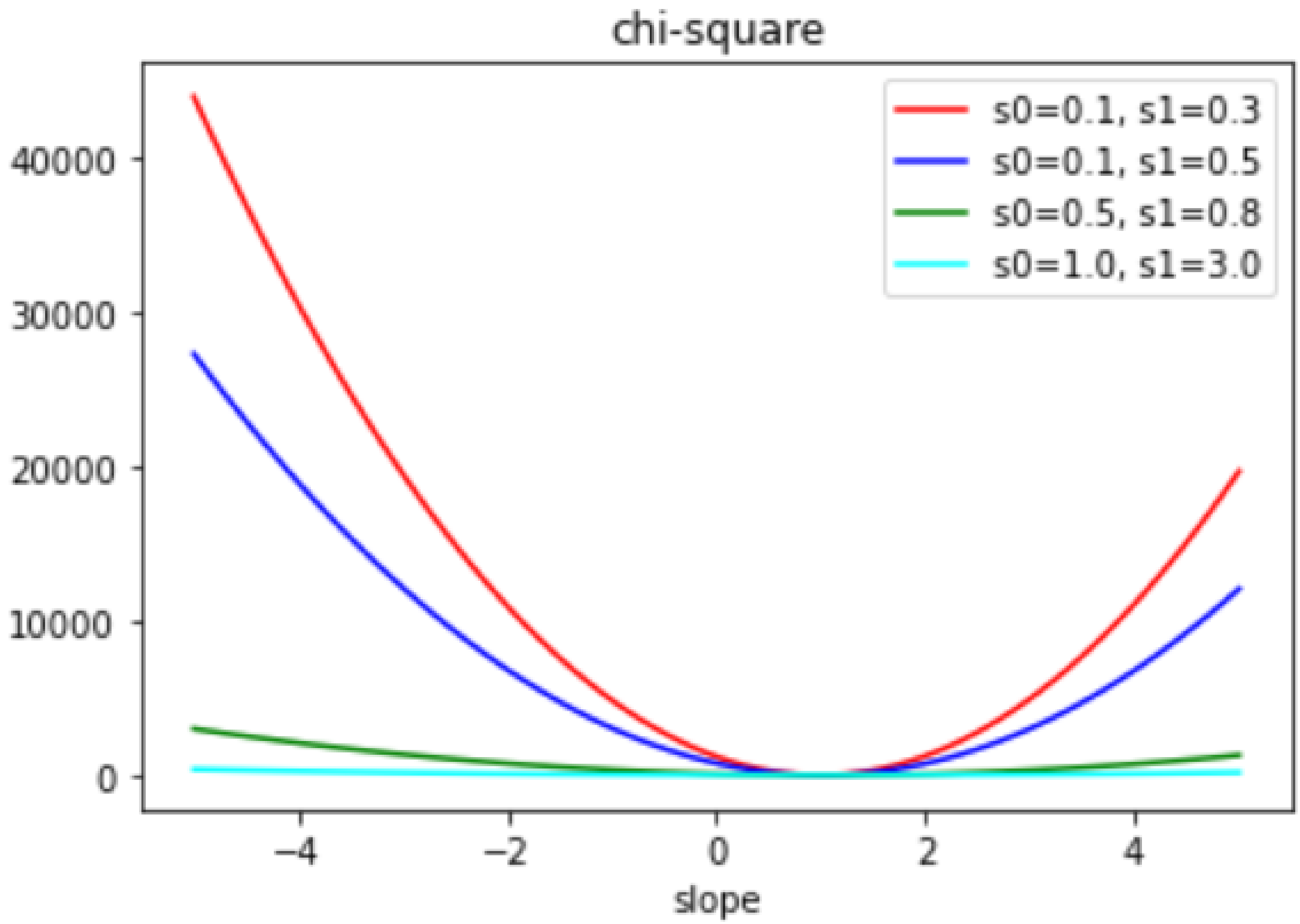



```
chi2s4, scales4 = fisher(1.0, 1.0, 3.0)
best fit a =  0.8566797062723266 , b =  0.03459640049619748
```

Comparison

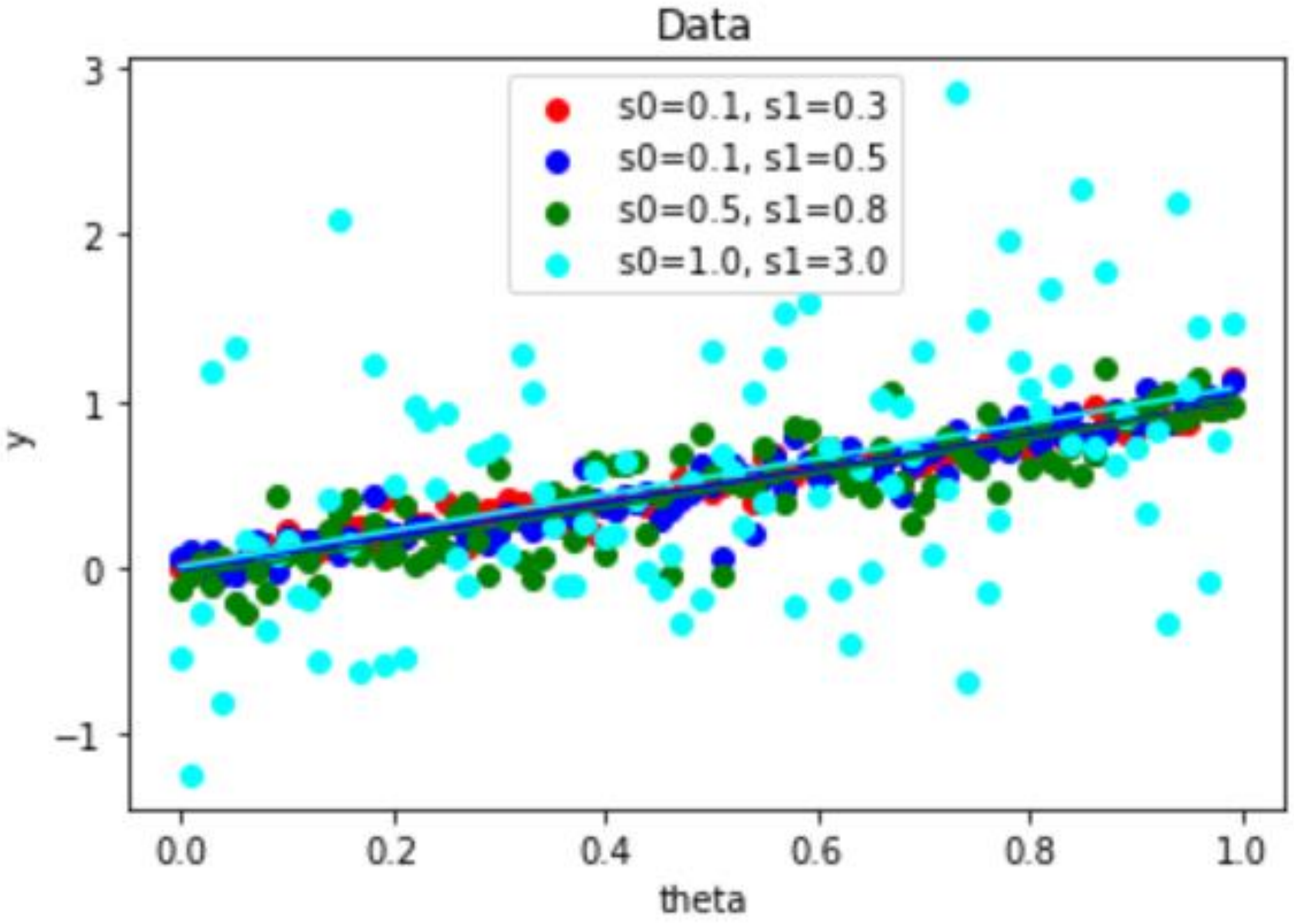
```
plt.plot(scales1, chi2s1, label="s0=0.1, s1=0.3", color="red")
plt.plot(scales1, chi2s2, label="s0=0.1, s1=0.5", color="blue")
plt.plot(scales1, chi2s3, label="s0=0.5, s1=0.8", color="green")
plt.plot(scales1, chi2s4, label="s0=1.0, s1=3.0", color="cyan")
plt.xlabel("slope")
plt.title("chi-square")
plt.legend()
```

- Fisher matrix value is curvature of chi-square curves



Comparison

```
plt.xlabel("theta")
plt.ylabel("y")
plt.title("Data")
plt.scatter(theta1, data1, label="s0=0.1, s1=0.3", color="red")
plt.plot(theta1, y1, color="red")
plt.scatter(theta1, data2, label="s0=0.1, s1=0.5", color="blue")
plt.plot(theta1, y2, color="blue")
plt.scatter(theta1, data3, label="s0=0.5, s1=0.8", color="green")
plt.plot(theta1, y3, color="green")
plt.scatter(theta1, data4, label="s0=1.0, s1=3.0", color="cyan")
plt.plot(theta1, y4, color="cyan")
plt.legend()
plt.show()
```



Fisher Matrix

- $F_{jk} = -\left\langle \frac{\partial^2 \ln p(M|D)}{\partial\theta_j \partial\theta_k} \right\rangle$
- Cramer-Rao lower bound
- $\langle \delta\theta_j \delta\theta_k \rangle \geq (F^{-1})_{jk}$ it is same for Gaussian

χ^2 distribution

χ^2 distribution

- Let x is a random number of standard normal distribution
- pdf of x is $\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$
- Suppose that $y = x^2$
- What is the pdf of y?

- Let $y = g(x)$ and pdf of x is $f_X(x)$
- How to find pdf of y, $f_Y(y)$

χ^2 distribution

- $x = g^{-1}(y)$, since $y = g(x)$ monotonic function
- $dx = \frac{dg^{-1}(y)}{dy} dy$
- Distribution of x is $f_X(x)dx$
- $f_X(x)dx = f_X(g^{-1}(y))\frac{dg^{-1}(y)}{dy}dy = f_Y(y)dy$
- Hence $f_Y(y) = f_X(g^{-1}(y))\frac{dg^{-1}(y)}{dy}$
- If $g(x) = x^2$, then $x = g^{-1}(y) = \pm\sqrt{y}$, $y \geq 0$
- $f_Y(y) = 2f_X(g^{-1}(y))\frac{dg^{-1}(y)}{dy} = \frac{2}{\sqrt{2\pi}}e^{-\frac{y}{2}}\frac{d\sqrt{y}}{dy} = \frac{1}{\sqrt{2}\sqrt{\pi}}y^{-\frac{1}{2}}e^{-\frac{y}{2}}$, χ_1^2 distribution

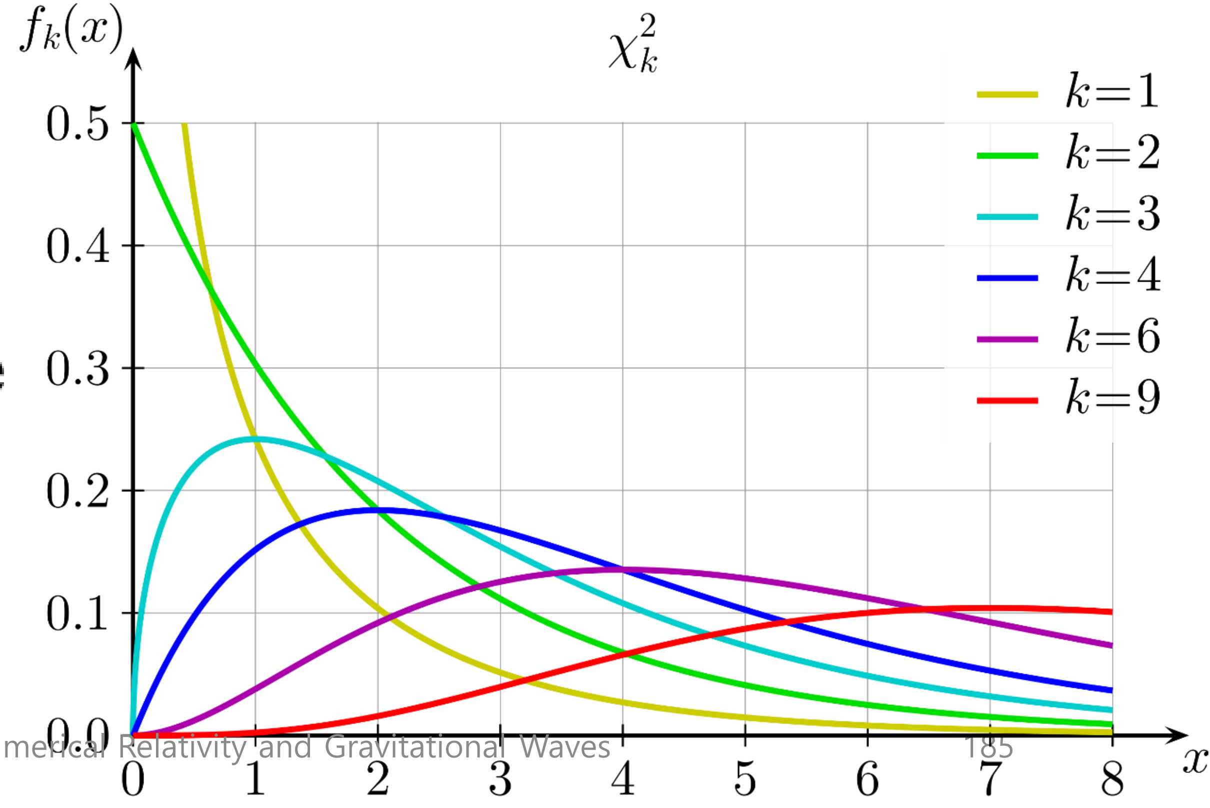
χ^2 distribution

- $y = x^2$ and $x \sim N(0,1)$
- $y < 0$, $P(Y < y) = 0$
- $y \geq 0$, $P(Y < y) = P(X^2 < y) = P(|X| < y) = P\left(-\sqrt{y} < X < \sqrt{y}\right) = F_X\left(\sqrt{y}\right) - F_X\left(-\sqrt{y}\right) = F_X\left(\sqrt{y}\right) - \left(1 - F_X\left(\sqrt{y}\right)\right) = 2F_X\left(\sqrt{y}\right) - 1$
- $f_Y(y) = \frac{d}{dy}P(Y < y) = 2\frac{d}{dy}F_X\left(\sqrt{y}\right) = 2\frac{d}{dy}\int_{-\infty}^{\sqrt{y}}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}dt = \frac{2}{\sqrt{2\pi}}e^{-\frac{y}{2}}\frac{d\sqrt{y}}{dy} = \frac{1}{\sqrt{2}\sqrt{\pi}}y^{-\frac{1}{2}}e^{-\frac{y}{2}} = \frac{1}{2^{1/2}\Gamma\left(\frac{1}{2}\right)}y^{-\frac{1}{2}}e^{-\frac{y}{2}}$

χ^2 distribution

- $y_i = x_i^2$, $Q = \sum_{i=1}^{k} y_i = \sum_{i=1}^{k} x_i^2$, $x_i \sim N(0,1)$
- pdf of Q is χ_k^2, χ^2 distribution with k degrees of freedom

$$f(x;\,k) = \begin{cases} \dfrac{x^{\frac{k}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{k}{2}}\,\Gamma\left(\frac{k}{2}\right)}, & x > 0; \\ 0, & \text{otherwise} \end{cases}$$

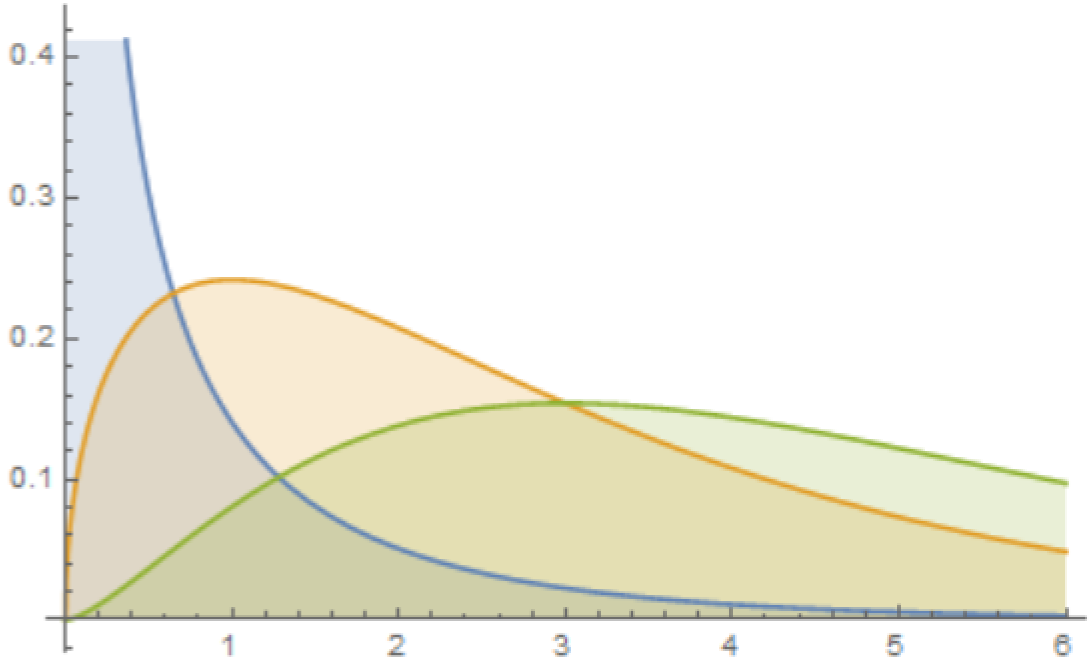


[Chi-square distribution - Wikipedia]

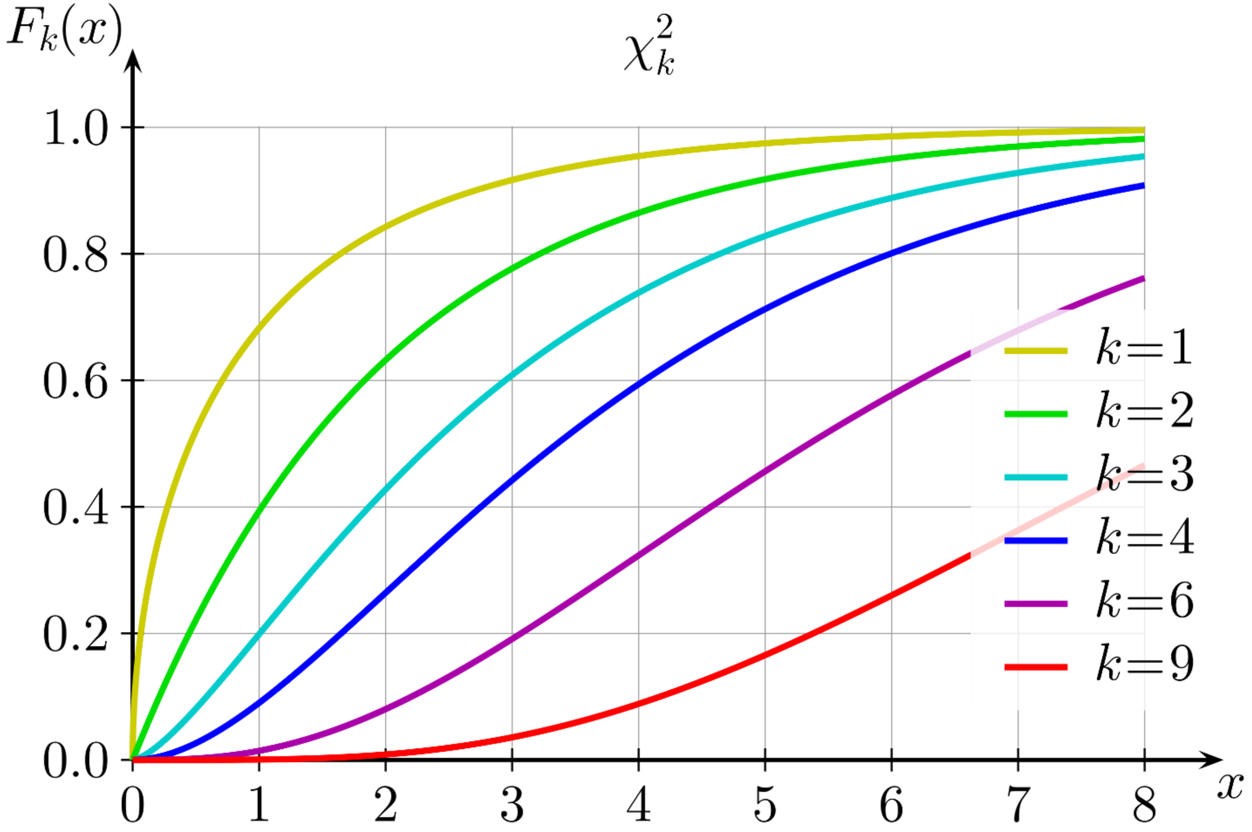
χ^2 distribution

```
In[22]:= Plot[Table[PDF[ChiSquareDistribution[ν], x], {ν, {0.5, 3, 5}}] // Evaluate,
         |플롯  |목록…  |…     |카이 제곱 분포                                   |평가
          {x, 0, 6}, Filling → Axis]
                     |채우기      |축을 그릴지 여부
```

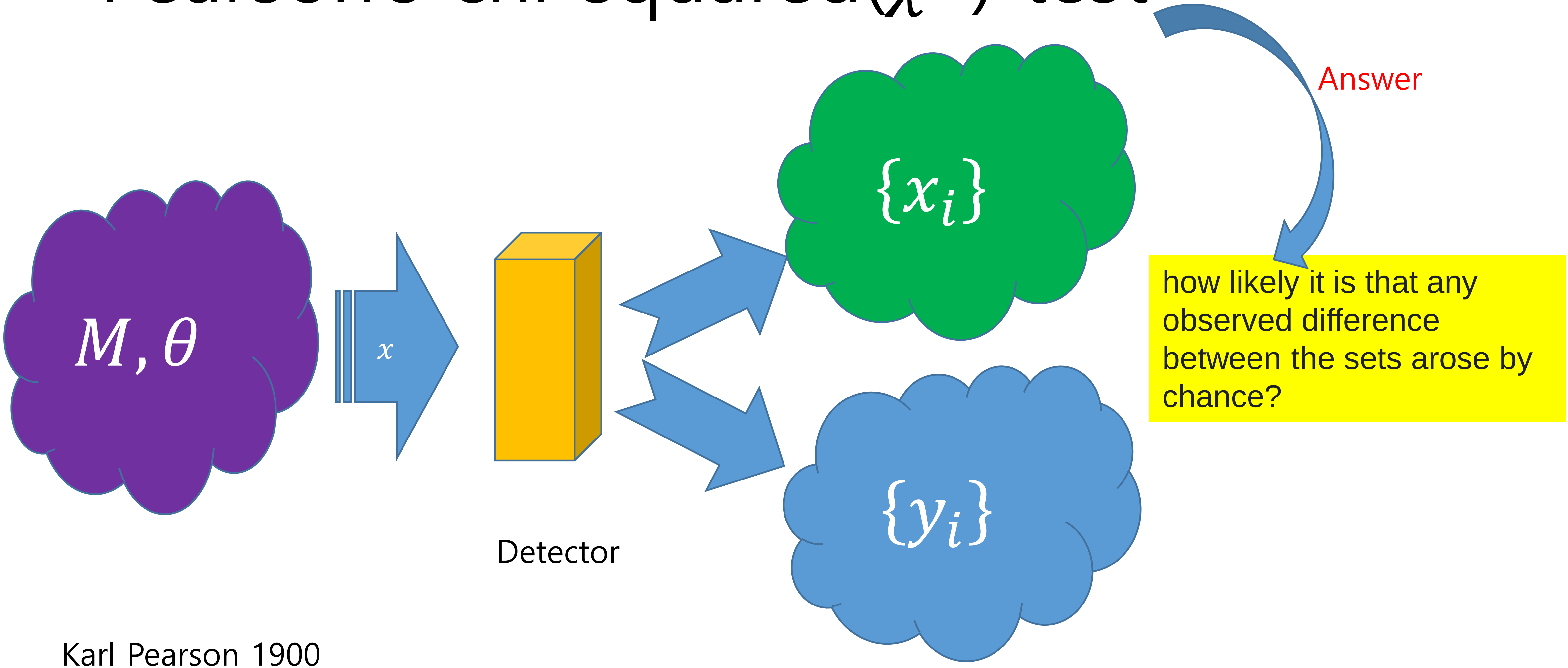
Out[22]=



[Chi-square distribution - Wikipedia]



Pearson's chi-squared(χ^2) test



Karl Pearson 1900

Pearson's chi-squared(χ^2) test process

1. Calculate χ^2, sum of squared deviation between observed and theoretical values
2. Determine the degrees of freedom(dof)
   1. Goodness-of-fit : number of observations – number of parameters
3. Select level of confidence
4. Compare χ^2 to critical value for given dof and confidence level
5. Sustain or reject the null hypothesis. $\chi^2 > \text{critical}$: reject H_0, $\chi^2 < \text{critical}$: sustain H_0, but not necessarily accepted

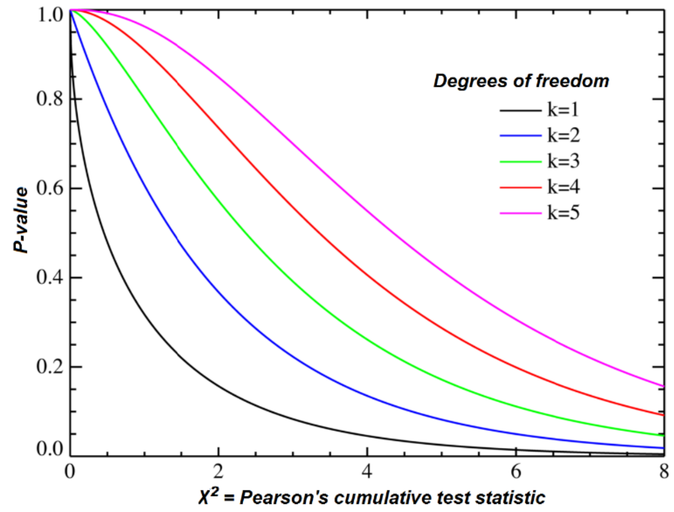
Pearson's chi-squared(χ^2) test process

M, θ

x

Detector

$\{x_i\}$



[국가슈퍼컴퓨팅센터 (ksc.re.kr)](ksc.re.kr)

$\{y_i\}$

Theoretical expectation

$\chi^2, \nu, \chi_c^2, p, \alpha$

$\chi^2 > \chi_c^2$ reject H_0 Data is not describing the theory

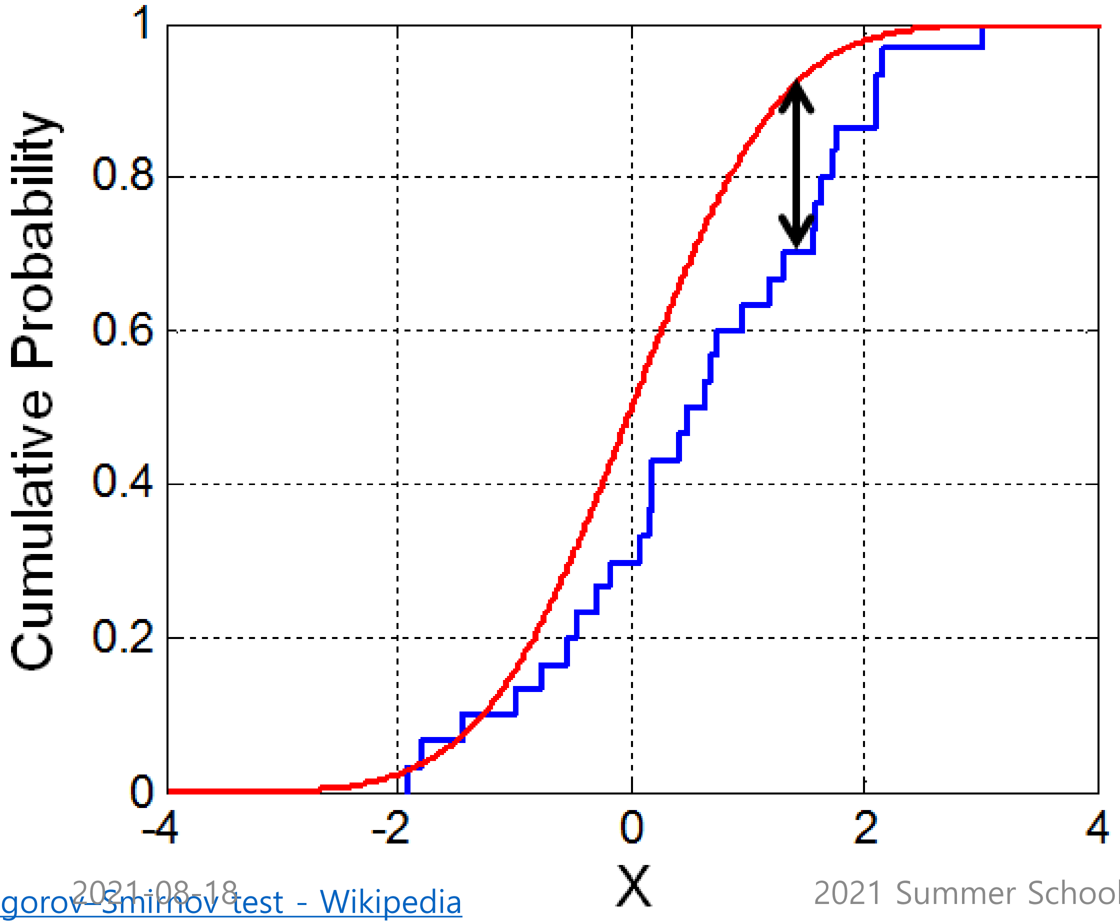
$\chi^2 < \chi_c^2$ sustain H_0 Data is describing the theory but not definitely

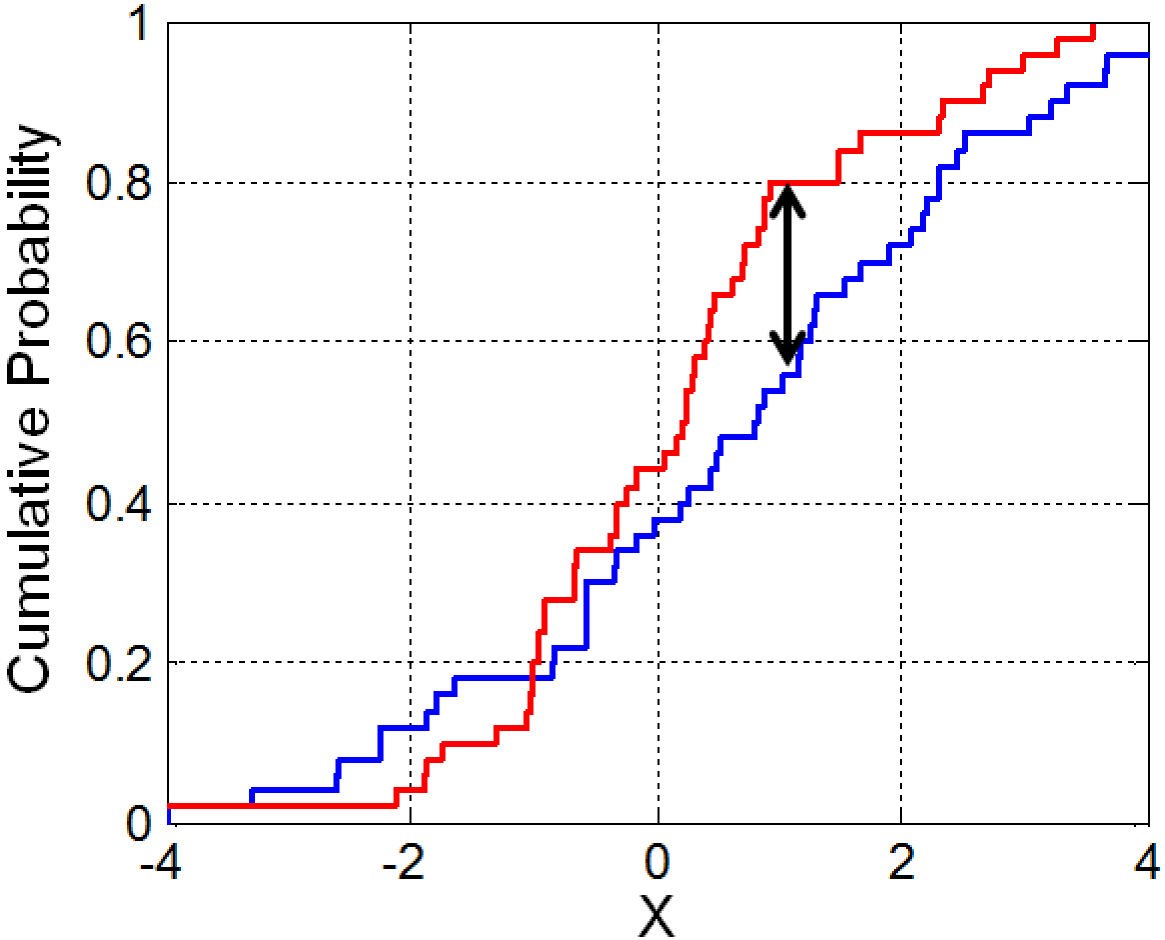
Pearson's chi-squared(χ^2) test

- $\chi^2 = \sum \frac{\left(y_i - y(x_i)\right)^2}{\sigma_i^2}$
- $\chi_\nu^2 = \frac{\chi^2}{\nu} = \frac{s^2}{\langle \sigma_i^2 \rangle}$
- $s^2 = \frac{1}{N-m} \sum w_i \left(y_i - y(x_i)\right)^2$
- $w_i = \frac{1/\sigma_i^2}{(1/N) \sum\left(1/\sigma_i^2\right)}$

Other kind of test

- Kolmogorov-Smirnov test





[Kolmogorov–Smirnov test - Wikipedia]

Other kind of test

- Anderson-Darling test
- Kuiper's test
- Shapiro-Wilk test
- Jarque-Bera test
- Goodness of fit

References

- Interferometer techniques for gravitational-wave detection (nih.gov) (Living Rev. Relativ. 2017 August 15)
- Maggiore, "Gravitational Waves: Volume 1, Theory and Experiments"
- Creighton & Anderson, "Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis"
- P. Jaranowski, A. Kerolak, "Analysys of Gravitational-Wave Data", Cambridge University Press, 2009
- J.T. Whelan, "The Geometry of Gravitational Wave Detection", T1300666

Thanks!
Questions?