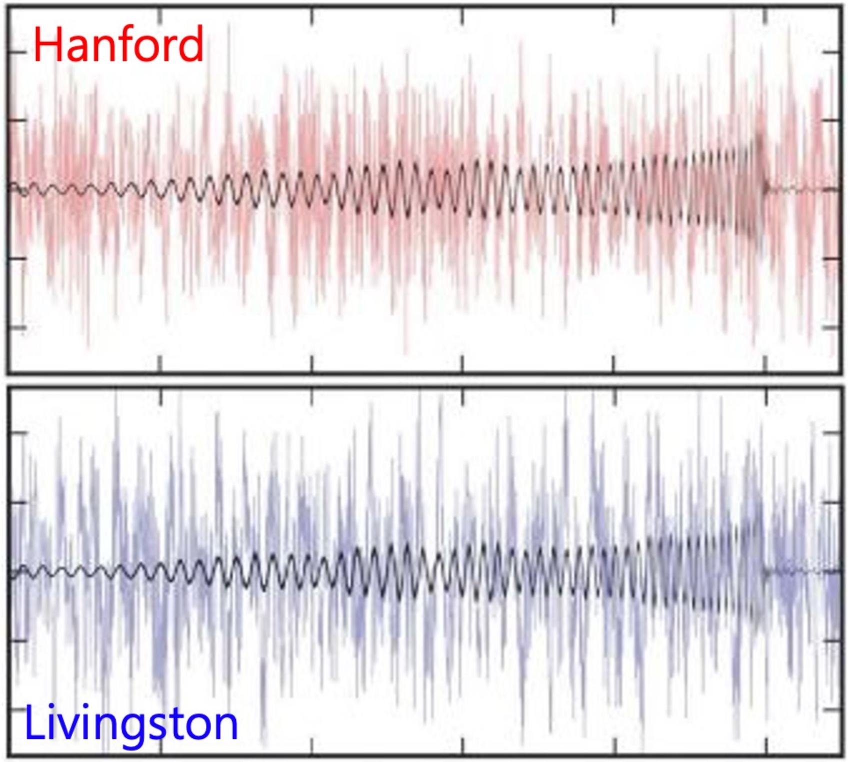
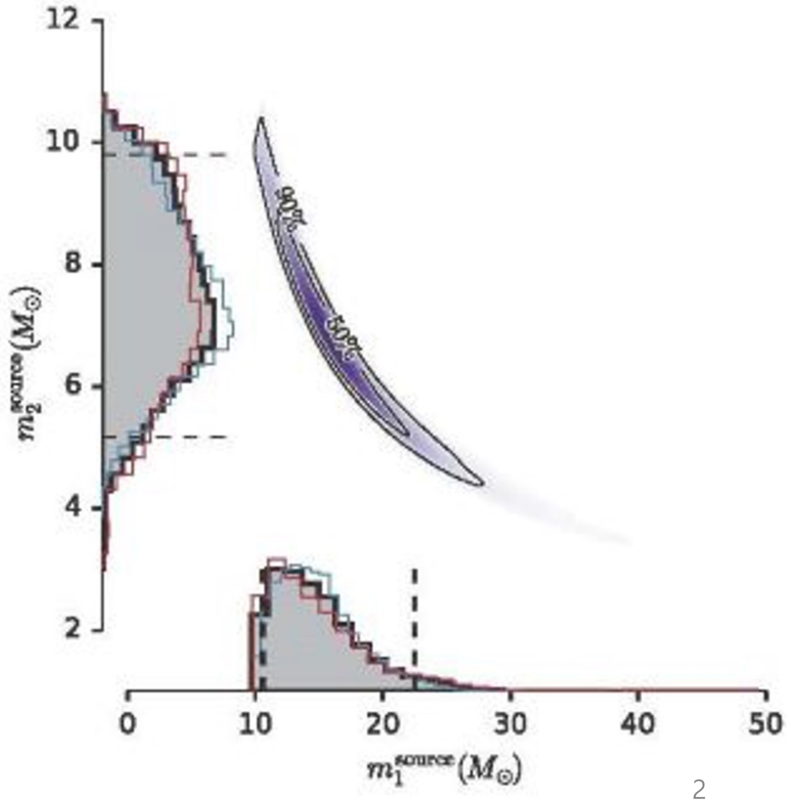
중력파 모수추정

JeongCho Kim(SNU)

2024 GWNR Winter School
2024.01.31

Parameter Estimation (PE)





Frequentist vs Bayesian

Frequentist Statistics

- all about probability in the long run

- frequentist analyses base their results onl
y on the data they collect

Bayesian Statistics

- describes a method to update probabilities based on data and past knowledge

- parameters and hypotheses are seen as pro
bability distributions and the data as fixed

The prior is one of the key differences between frequentist and Bayesian inference

Bayesian inference

Bayesian inference is a way of making statistical inferences in which the statistician assigns prior distribution to the distributions that could generate the data.

After the data is observed, Bayes' rule is used to update the prior, that is, posterior distribu tion assigned to the possible data generating distributions.

- We observe some data (a sample), that we collect in a vector x.
- We regard x as the realization of a random vector X.
- We do not know the probability distribution of X.
- We define a statistical model, that is a set θ of probability distributions that could have generated the data.
- We parametrize the model, that is, we put the elements of θ in correspondence with a set of real vectors called parameters
- we use the sample and the statistical model to make an inference about the unknown data generating distribution

Bayes' rule

It is a rule for computing conditional probabilities.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Let be two events, A and B.

Denote it's probabilities by P(A) and P(B).

Suppose the both P(A)>0 and P(B).

Bayes' rule – proof

conditional probability formula : $$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad P(B|A) = \frac{P(A \cap B)}{P(A)}$$

The second formula re-arranged as follows : $$P(B|A)P(A) = P(A \cap B)$$

It is plugged into the first formula, we obtain : $$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes theorem

Initial Understanding + New Observation = Updated Understanding

D : the observed data
θ : the model parameters

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} = \frac{P(D|\theta)P(\theta)}{\int P(\theta)P(D|\theta)\ d\theta}$$

Posterior = Likelihood * Prior

The likelihood

The first building block of a parametric Bayesian model is the likelihood $p(D|\theta)$

Suppose that the sample is a vector of n independent and identically distributed draws $x_1, \cdots, x_n$ from a normal distribution.

$$x = [x_1 \cdots x_n]$$

The mean (μ) of the distribution is unknown, while its variance σ^2 is known. These are the two parame ters of the model.

The Probability density function of a generic draw x_i is $P(x_i|\mu) = (2\pi\sigma^2)^{-1/2} exp\left(-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\right)$,

where we use the notation $P(x_i|\mu)$ to highlight the fact that μ is unknow and the density of x_i depen ds on this unknow parameter.

Because the observation $x_1 \cdots x_n$ are independent, we can write the likelihood as

$$P(x|\mu) = \prod_{i=1}^{n} P(x_i|\mu) = (2\pi\sigma^2)^{-n/2} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right),$$

The likelihood

$p(D|\theta)$: The probability that the gravitational wave generated by the model parameter θ produces the signal D.

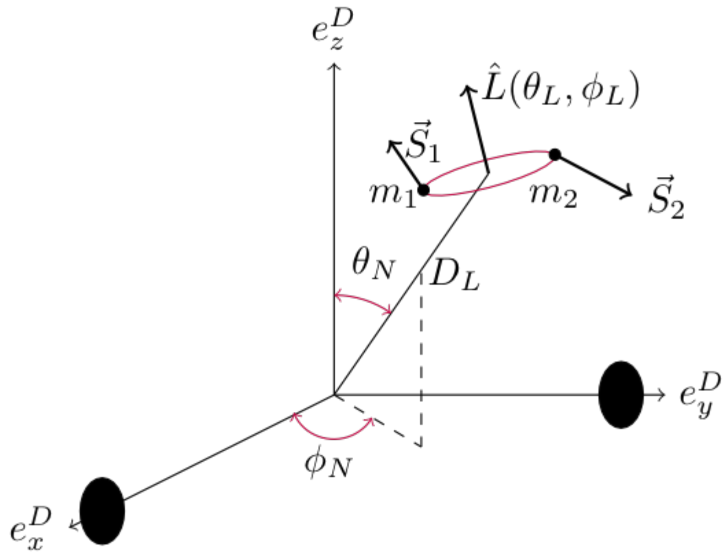
$$P(D|\theta) = \prod_{i=1}^{n} P(d_i|\theta) = (2\pi\sigma^2)^{-n/2} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(d_i-\theta)^2\right) \propto e^{<D-h|D-h>/2}$$

$$\langle a|b\rangle \equiv 4\Re\int_0^\infty \frac{\tilde{a}(f)\tilde{b}^*(f)}{S(f)}df$$

$$\langle a|b\rangle = 2\Re\int_{-\infty}^{\infty} \frac{\tilde{a}(f)\tilde{b}^*(f)}{S(|f|)}df = \int_{-\infty}^{\infty} \frac{\tilde{a}(f)\tilde{b}^*(f)+\tilde{a}^*(f)\tilde{b}(f)}{S(|f|)}df$$

The prior

The prior is the subjective probability density assigned to the parameter θ.



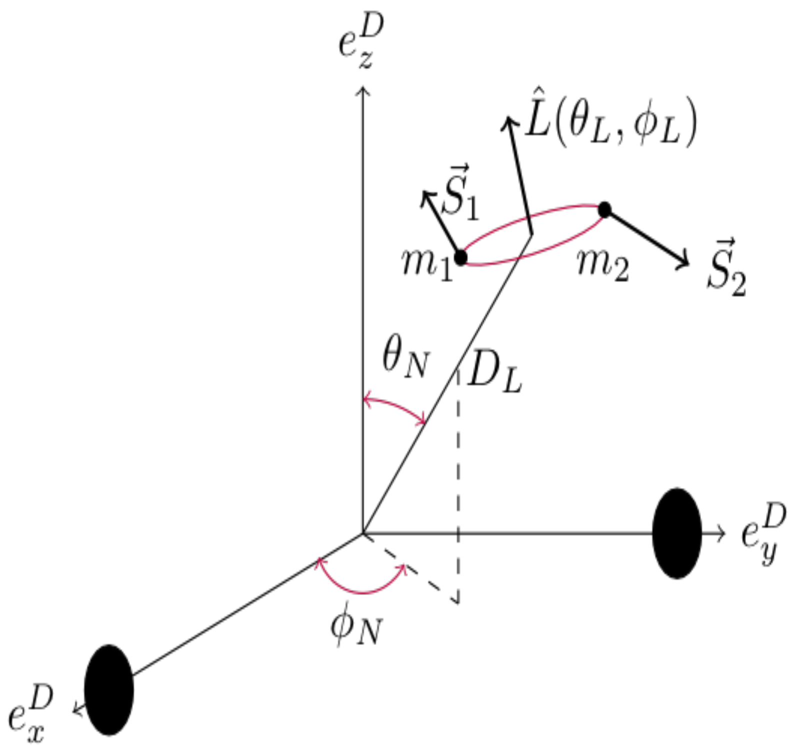
$$p(h') = \delta(h' - h'(\vec{\theta}))p(\vec{\theta})$$

priors on the system parameters

What we know about the parameters of the system *before* obtaining the data

Intrinsic & extrinsic parameters

m_1 m_2	Uniform in some range
$\vec{S}_2$ $\vec{S}_1$	Uniform in direction and magnitude in $[0, m_i^2]$
λ_1 λ_2	Uniform in (0,5000)



D_L	Uniform in volume
θ_N ϕ_N	Uniform in the sky
θ_L ϕ_L	Uniform in direction

The posterior

The posterior gives the probability density that a model describes the data.

$P(\theta|D)$: The posterior probability is the conditional probability $P(\theta|D)$, calculated after receiving the information that the event D has happened.

The marginal distribution $f(x)$ is derived from the prior and the likelihood.

We first derive the joint distribution $f(x,\theta) = f(x|\theta)f(\theta)$ and the we marginalize it to obtain the posterior.

In the continuous case, the marginal is computed by integration as $f(x) = \int f(x,\theta)d\theta$.

In the discrete case, it is derived by calculating a sum as $f(x) = \sum_{\theta} f(x,\theta)$.

→ there are numerical methods that allow us to draw MCMC samples from the posterior distribution.

Probability density function (PDF)

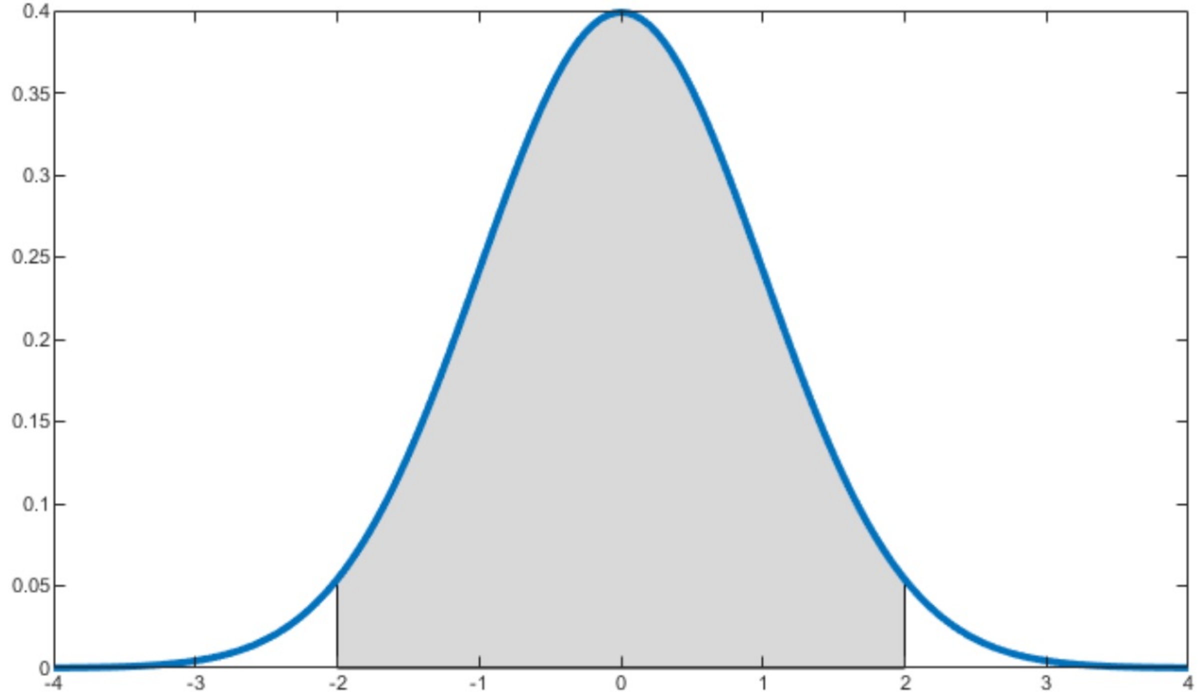
The probability density function is a function that completely characterizes the distribution of a continuous random variable.

The PDF of a continuous random variable X is a function $PDF_X\colon \mathbb{R} \to [0, \infty)$ such that

$$P(X \in [a, b]) = \int_a^b PDF_X(x)\, dx$$

for any interval $[a, b] \subseteq \mathbb{R}$.

The set of values x for which $PDF_X(x) > 0$ is called the support of X.



Markov Chain Monte Carlo (MCMC)

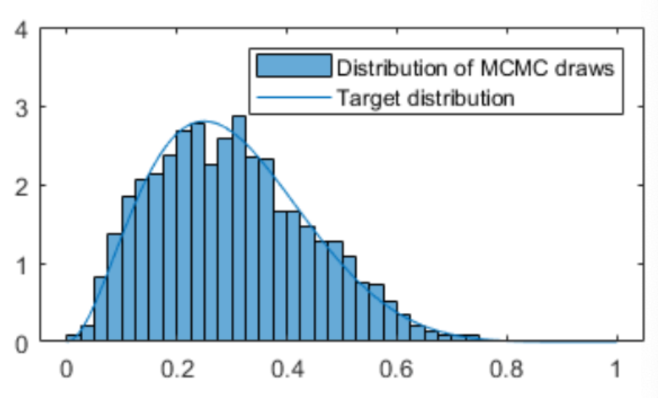
Markov Chain Monte Carlo (MCMC) methods are very powerful Monte Carlo methods that are often used in Bayesian inference.

The classical Monte Carlo methods rely on computer-generated samples made up of independent o bservations.

MCMC methods are used to generate sequences of dependent observations. These sequences are Markov chains.

→ MCMC methods work like standard Monte Carlo methods, although with a twist: the comput er-generated draws $x_1, \cdots, x_2$are not independent, but they are serially correlated.
→ They are the relizations of T random variables $X, \cdots, X_T$ that form a Markov Chain.

Markov property



A random sequence (X_t) is a Markov chain if and only if, given the current value X_t, the future obser vation X_{t+n} are conditionally independent of the past values X_{t-k}, for any positive integers k and n.

$$P(X_{t+n} = x|X_t, X_{t-1}, \cdots, X_{t-k}) = P(X_{t+n} = x|X_t)$$

This property, known as the Markov property, The probability distribution of the future values of the chain depends only on its current value X_t, regardless of how the value was reached regardless of th e path followed by the chain in the past

Although this is not true in general of any Markov chain, the chains generated by MCMC meth od have the following property :

Two variables X_t and X_{t+n} are not independent, but they become closer and closer to being ind ependent as n increases.
This property implies that $f(X_{t+n}|X_t)$ converges to $f(X_{t+n})$ as n becomes large.

Bayes theorem

Initial Understanding + New Observation = Updated Understanding

D : the observed data
θ : the model parameters

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} = \frac{P(D|\theta)P(\theta)}{\int P(\theta)P(D|\theta)\ d\theta}$$

The evidence

Normalization factor for parameter estimation

Important for model selection

Compare competing models, for example
'GW170817 was a BNS' vs 'GW170817 was a BBH'

$$\mathcal{O}_{ij} = \frac{p(M_i|d)}{p(M_j|d)}$$

$$= \frac{p(M_i)p(d|M_i)}{p(M_j)p(d|M_j)}$$

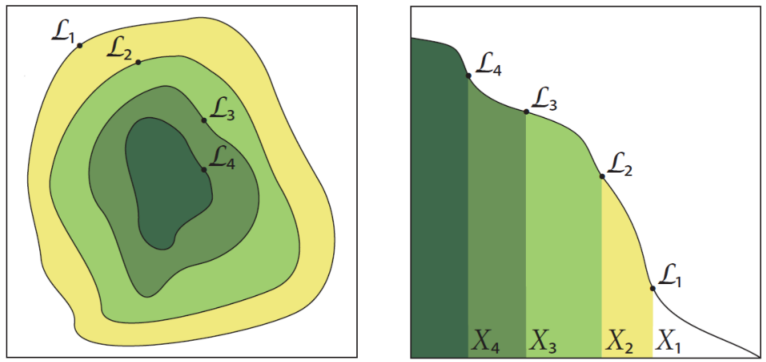
Bayes Factor
Likelihood of the models
Ratio of the evidences

Nested Sampling

Nested sampling estimates directly how the likelihood function relates to prior mass. The evidence (alternatively the marginal likelihood, marginal density of the data, or the prior predictive) is immediately obtained by summation.

$$Z = \text{evidence} = \int L\, dX$$

$L = L(\theta)$ is the likelihood function.
$dX = \pi(\theta)d\theta$ is the element of prior mass and θ represents the unknown parameters.



Feroz et al. (2013)

Setting the priors

The model describing the black hole merger depends on 15 parameters. Running a sampler with this many parameters can take half a day to run. To save some time, I'll fix all but 4 parameters: the chirp mass, the mass ratio, the phase, and the geocent time.

The chirp mass and mass ratio are two parameters which depend only on the mass of the two black holes that caused the gravitational waves. When these two parameters are derived, we can use them to calculate the masses of the black holes. Since we don't know much about the system parameters that we're allowing to vary, uniform priors are most appropriate:

```
prior = bilby.core.prior.PriorDict()
# uniform priors for variable parameters
prior['chirp_mass'] = Uniform(name='chirp_mass', minimum=10.0,maximum=100.0)
prior['mass_ratio'] = Uniform(name='mass_ratio', minimum=0.5, maximum=1)
prior['phase'] = Uniform(name="phase", minimum=0, maximum=2*np.pi)
prior['geocent_time'] = Uniform(name="geocent_time", minimum=event_start_time-0.
                                maximum=event_start_time+0.1)
# fixed values for all other parameters
prior['a_1'] =  0.0
prior['a_2'] =  0.0
prior['tilt_1'] =  0.0
prior['tilt_2'] =  0.0
prior['phi_12'] =  0.0
prior['phi_jl'] =  0.0
prior['dec'] =  -1.2232
prior['ra'] =  2.19432
prior['theta_jn'] =  1.89694
prior['psi'] =  0.532268
prior['luminosity_distance'] = 412.066
```

Setting the likelihood

Bilby comes with in-built likelihood functions. To use them, first a waveform has to be generated using the properties of the data, like duration and sampling frequency. We also need to combine the interferometers into a list, so that both will be used at once when it comes to sampling.

```
interferometers = [H1, L1]

waveform_arguments = dict(waveform_approximant='IMRPhenomPv2',
                          reference_frequency=50., minimum_frequency=20.)

# set sampling frequency of data
sampling_frequency = 2048.

# generate waveforms
waveform_generator = bilby.gw.WaveformGenerator(
    duration=duration, sampling_frequency=sampling_frequency,
    frequency_domain_source_model=bilby.gw.source.lal_binary_black_hole,
    parameter_conversion=bilby.gw.conversion.convert_to_lal_binary_black_hole_pa
    waveform_arguments=waveform_arguments)

likelihood = bilby.gw.GravitationalWaveTransient(
    interferometers=interferometers, waveform_generator=waveform_generator)
```

Sampling the data

Now that the likelihood and prior functions are set up, the sampler can now be implemented to derive some of the properties of the binary black hole system. This process is intensive, and takes a long time (1 hour 15 minutes) to run on my machine.

The sampler I chose to use is "dynesty", which uses nested sampling to estimate the model parameters. For a more in-depth work through of dynesty, you can read through an example of gamma-ray spectroscopy using dynesty on this site.

Using bilby, we can run the sampler. First, we have to set the number of live points, and the stopping criterion as hyperparameters. For neatness, I've removed the log that bilby outputs whilst running. Instead, I'll just show the summary that is outputted by bilby once the sampler has finished iterating. This summary contains information on the log evidence, and the Bayes factor of the model.

```
nlive = 1000          # live points
stop = 0.1            # stopping criterion
method = "unif"       # method of sampling
sampler = "dynesty"   # sampler to use

result = bilby.run_sampler(
    likelihood, prior, sampler=sampler, outdir=outdir, label=label,
    conversion_function=bilby.gw.conversion.generate_all_bbh_parameters,
    sample=method, nlive=nlive, dlogz=stop)
```

GW - Q.4

$conda deactivate
$conda env list
$conda activate igwn-py39

$nohup python *.py

$tail –f nohup.out

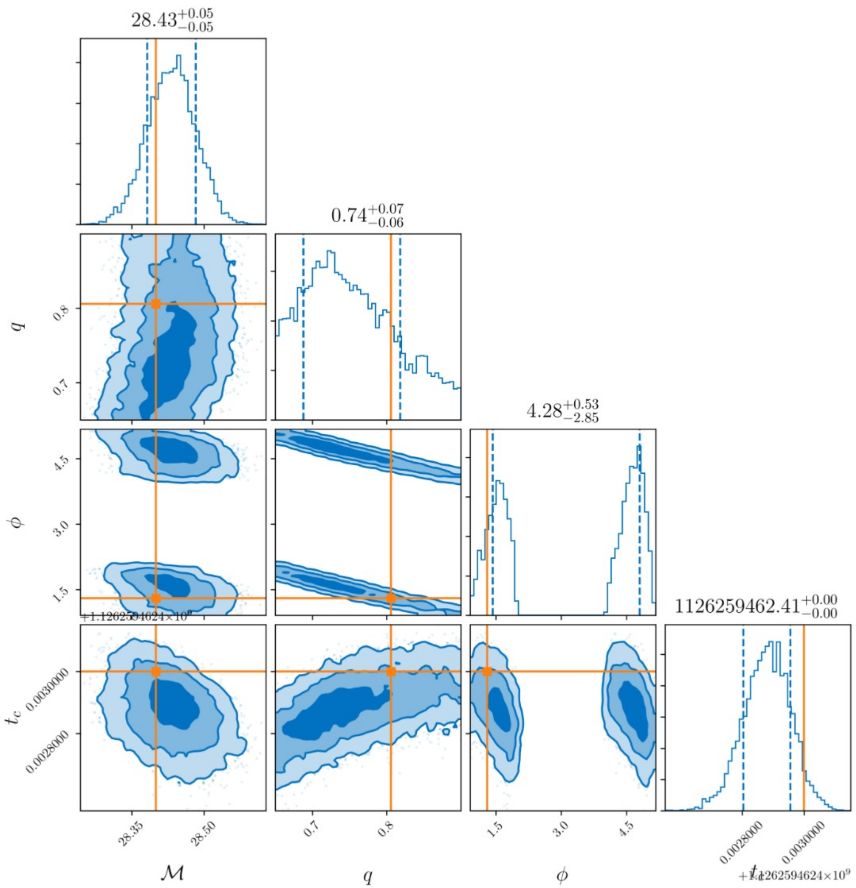
```
(igwn-py39) [nrgw20@olaf-c197 ~]$ nohup: ignoring input and appending output to 'nohup.out'
(igwn-py39) [nrgw20@olaf-c197 ~]$ tail -f nohup.out
```

```
^M1it [00:00, ?it/s]^M17it [00:00, 158.74it/s, bound:0 nc:  1 ncall:2.7e+02 eff:6.4% logz-ratio=-637.48+/-0.16 dlogz:728.129>1]^M33it [00:00, 1
59.27it/s, bound:0 nc:  1 ncall:2.8e+02 eff:11.6% logz-ratio=-406.13+/-0.16 dlogz:499.249>1]^M49it [00:00, 144.74it/s, bound:0 nc:  1 ncall:3.0
e+02 eff:16.1% logz-ratio=-331.38+/-0.15 dlogz:420.916>1]^M64it [00:00, 132.85it/s, bound:0 nc:  1 ncall:3.3e+02 eff:19.6% logz-ratio=-294.66+/
-0.15 dlogz:382.884>1]^M78it [00:00, 128.06it/s, bound:0 nc:  1 ncall:3.5e+02 eff:22.5% logz-ratio=-261.97+/-0.14 dlogz:349.704>1]^M91it [00:00
, 116.82it/s, bound:0 nc:  3 ncall:3.7e+02 eff:24.6% logz-ratio=-223.03+/-0.16 dlogz:315.152>1]^M103it [00:00, 114.27it/s, bound:0 nc:  2 ncall
:3.9e+02 eff:26.5% logz-ratio=-199.38+/-0.14 dlogz:286.521>1]^M116it [00:00, 115.02it/s, bound:0 nc:  4 ncall:4.1e+02 eff:28.4% logz-ratio=-183
.28+/-0.14 dlogz:270.368>1]^M130it [00:01, 120.19it/s, bound:0 nc:  1 ncall:4.3e+02 eff:30.5% logz-ratio=-164.19+/-0.13 dlogz:251.156>1]^M143it
 [00:01, 101.75it/s, bound:0 nc:  1 ncall:4.6e+02 eff:31.4% logz-ratio=-152.98+/-0.14 dlogz:240.178>1]^M156it [00:01, 103.14it/s, bound:0 nc:
4 ncall:4.8e+02 eff:32.7% logz-ratio=-141.53+/-0.13 dlogz:228.421>1]^M167it [00:01, 90.74it/s, bound:0 nc:  1 ncall:5.0e+02 eff:33.1% logz-rati
o=-132.70+/-0.14 dlogz:219.455>1] ^M178it [00:01, 91.84it/s, bound:0 nc:  3 ncall:5.2e+02 eff:33.9% logz-ratio=-126.86+/-0.14 dlogz:213.818>1]^
M188it [00:01, 88.65it/s, bound:0 nc:  2 ncall:5.5e+02 eff:34.4% logz-ratio=-119.41+/-0.14 dlogz:206.273>1]^M198it [00:01, 79.20it/s, bound:0 n
c:  1 ncall:5.7e+02 eff:34.5% logz-ratio=-107.48+/-0.15 dlogz:194.950>1]^M207it [00:01, 80.98it/s, bound:0 nc:  1 ncall:5.9e+02 eff:35.0% logz-
ratio=-100.15+/-0.14 dlogz:187.004>1]^M216it [00:02, 71.26it/s, bound:0 nc:  1 ncall:6.2e+02 eff:34.8% logz-ratio=-94.93+/-0.14 dlogz:181.469>1
] ^M224it [00:02, 68.95it/s, bound:0 nc:  8 ncall:6.4e+02 eff:34.8% logz-ratio=-89.85+/-0.14 dlogz:176.616>1]^M232it [00:02, 69.93it/s, bound:0
 nc:  3 ncall:6.6e+02 eff:35.0% logz-ratio=-86.65+/-0.13 dlogz:219.964>1]^M240it [00:02, 70.72it/s, bound:0 nc:  1 ncall:6.8e+02 eff:35.2% logz
-ratio=-83.96+/-0.13 dlogz:217.333>1]^M248it [00:02, 71.18it/s, bound:0 nc:  5 ncall:7.0e+02 eff:35.4% logz-ratio=-77.74+/-0.14 dlogz:211.152>1
]^M256it [00:02, 69.57it/s, bound:0 nc:  3 ncall:7.2e+02 eff:35.5% logz-ratio=-74.53+/-0.14 dlogz:207.757>1]^M264it [00:02, 68.38it/s, bound:0
nc:  1 ncall:7.4e+02 eff:35.6% logz-ratio=-72.57+/-0.13 dlogz:205.684>1]^M271it [00:03, 56.94it/s, bound:0 nc:  4 ncall:7.7e+02 eff:35.1% logz-
ratio=-70.11+/-0.13 dlogz:203.236>1]^M280it [00:03, 60.80it/s, bound:0 nc:  7 ncall:8.0e+02 eff:35.2% logz-ratio=-67.33+/-0.13 dlogz:200.563>1]
^M287it [00:03, 61.53it/s, bound:0 nc:  6 ncall:8.1e+02 eff:35.3% logz-ratio=-65.15+/-0.13 dlogz:198.239>1]^M294it [00:03, 62.92it/s, bound:0 n
c:  1 ncall:8.3e+02 eff:35.3% logz-ratio=-62.37+/-0.14 dlogz:195.621>1]^M301it [00:03, 48.90it/s, bound:0 nc:  6 ncall:8.7e+02 eff:34.6% logz-r
atio=-60.11+/-0.14 dlogz:193.220>1]^M307it [00:03, 48.43it/s, bound:0 nc:  1 ncall:8.9e+02 eff:34.4% logz-ratio=-58.33+/-0.13 dlogz:191.280>1]^
M313it [00:03, 48.08it/s, bound:0 nc:  3 ncall:9.2e+02 eff:34.2% logz-ratio=-57.02+/-0.13 dlogz:190.029>1]^M319it [00:04, 43.96it/s, bound:0 nc
:  4 ncall:9.4e+02 eff:33.8% logz-ratio=-55.55+/-0.13 dlogz:188.456>1]^M325it [00:04, 45.98it/s, bound:0 nc:  6 ncall:9.6e+02 eff:33.7% logz-ra
tio=-53.90+/-0.13 dlogz:186.826>1]^M332it [00:04, 46.74it/s, bound:0 nc: 11 ncall:9.9e+02 eff:33.6% logz-ratio=-52.03+/-0.13 dlogz:184.922>1]^M
337it [00:04, 40.13it/s, bound:0 nc:  2 ncall:1.0e+03 eff:33.0% logz-ratio=-51.00+/-0.13 dlogz:183.794>1]^M343it [00:04, 42.61it/s, bound:0 nc:
  4 ncall:1.0e+03 eff:32.9% logz-ratio=-49.05+/-0.14 dlogz:181.948>1]^M348it [00:04, 42.32it/s, bound:0 nc:  2 ncall:1.1e+03 eff:32.8% logz-rat
io=-47.86+/-0.13 dlogz:180.612>1]^M354it [00:04, 45.53it/s, bound:0 nc:  3 ncall:1.1e+03 eff:32.7% logz-ratio=-46.62+/-0.13 dlogz:179.508>1]^M3
60it [00:04, 47.96it/s, bound:0 nc:  4 ncall:1.1e+03 eff:32.7% logz-ratio=-44.53+/-0.14 dlogz:177.322>1]^M366it [00:05, 49.80it/s, bound:0 nc:
 4 ncall:1.1e+03 eff:32.7% logz-ratio=-42.79+/-0.14 dlogz:175.581>1]^M372it [00:05, 35.49it/s, bound:0 nc:  1 ncall:1.2e+03 eff:31.8% logz-rati
o=-41.41+/-0.14 dlogz:174.121>1]^M377it [00:05, 36.85it/s, bound:0 nc:  2 ncall:1.2e+03 eff:31.7% logz-ratio=-40.40+/-0.13 dlogz:173.036>1]^M38
4it [00:05, 43.13it/s, bound:0 nc:  3 ncall:1.2e+03 eff:31.8% logz-ratio=-39.18+/-0.13 dlogz:171.795>1]^M391it [00:05, 44.77it/s, bound:0 nc:
8 ncall:1.2e+03 eff:31.7% logz-ratio=-38.01+/-0.13 dlogz:170.548>1]^M396it [00:05, 39.88it/s, bound:0 nc:  7 ncall:1.3e+03 eff:31.4% logz-ratio
=-36.94+/-0.13 dlogz:169.538>1]^M401it [00:06, 34.72it/s, bound:0 nc:  1 ncall:1.3e+03 eff:30.9% logz-ratio=-36.18+/-0.13 dlogz:168.670>1]^M407
it [00:06, 39.21it/s, bound:0 nc:  5 ncall:1.3e+03 eff:31.0% logz-ratio=-35.25+/-0.13 dlogz:167.744>1]^M412it [00:06, 38.36it/s, bound:0 nc:  2
 ncall:1.3e+03 eff:30.8% logz-ratio=-34.42+/-0.13 dlogz:166.896>1]^M418it [00:06, 42.35it/s, bound:0 nc:  2 ncall:1.4e+03 eff:30.8% logz-ratio=
-33.45+/-0.13 dlogz:165.941>1]^M423it [00:06, 41.58it/s, bound:0 nc:  4 ncall:1.4e+03 eff:30.7% logz-ratio=-32.66+/-0.13 dlogz:165.063>1]^M429i
t [00:06, 45.11it/s, bound:0 nc:  4 ncall:1.4e+03 eff:30.7% logz-ratio=-31.64+/-0.13 dlogz:164.097>1]^M434it [00:06, 38.70it/s, bound:0 nc:  4
ncall:1.4e+03 eff:30.3% logz-ratio=-30.57+/-0.14 dlogz:163.016>1]^M439it [00:06, 39.44it/s, bound:0 nc:  2 ncall:1.5e+03 eff:30.3% logz-ratio=-
29.72+/-0.13 dlogz:162.071>1]^M446it [00:07, 46.46it/s, bound:0 nc:  1 ncall:1.5e+03 eff:30.4% logz-ratio=-28.73+/-0.13 dlogz:161.069>1]^M452it
 [00:07, 46.16it/s, bound:0 nc:  9 ncall:1.5e+03 eff:30.3% logz-ratio=-27.87+/-0.13 dlogz:160.163>1]^M457it [00:07, 46.71it/s, bound:0 nc:  6 n
call:1.5e+03 eff:30.3% logz-ratio=-27.23+/-0.13 dlogz:159.465>1]^M462it [00:07, 40.51it/s, bound:0 nc:  1 ncall:1.5e+03 eff:30.0% logz-ratio=-2
6.77+/-0.13 dlogz:158.977>1]^M467it [00:07, 35.15it/s, bound:0 nc: 11 ncall:1.6e+03 eff:29.7% logz-ratio=-26.33+/-0.13 dlogz:158.507>1]^M471it
[00:07, 30.94it/s, bound:0 nc:  7 ncall:1.6e+03 eff:29.4% logz-ratio=-25.94+/-0.13 dlogz:158.110>1]^M477it [00:07, 32.88it/s, bound:0 nc: 13 nc
all:1.6e+03 eff:29.2% logz-ratio=-25.40+/-0.13 dlogz:157.542>1]^M481it [00:08, 25.53it/s, bound:0 nc:  1 ncall:1.7e+03 eff:28.7% logz-ratio=-25
.07+/-0.13 dlogz:157.182>1]^M484it [00:08, 21.35it/s, bound:0 nc:  8 ncall:1.7e+03 eff:28.2% logz-ratio=-24.78+/-0.13 dlogz:156.899>1]^M487it [
/dlogz                                                                                                                        85 2676 2702  98%
```

Plotting the posterior

Eventually, the sampling will be complete. Bilby creates a few useful plots of the posteriors, along with saving samples of the posteriors for us to further analyse. Lets start by plotting a corner plot, which nicely shows the posteriors for all the parameters, as well as contour plots showing how one parameter varies with any other.

```
result.plot_corner()
```



GW - Q.4

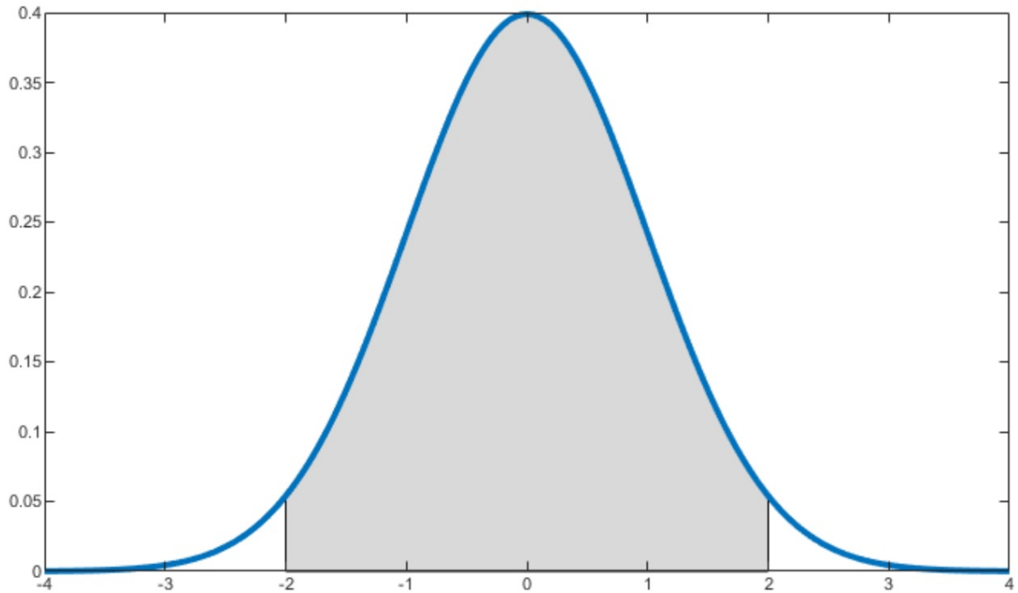
```
Mc = result_short.posterior["chirp_mass"].values
```

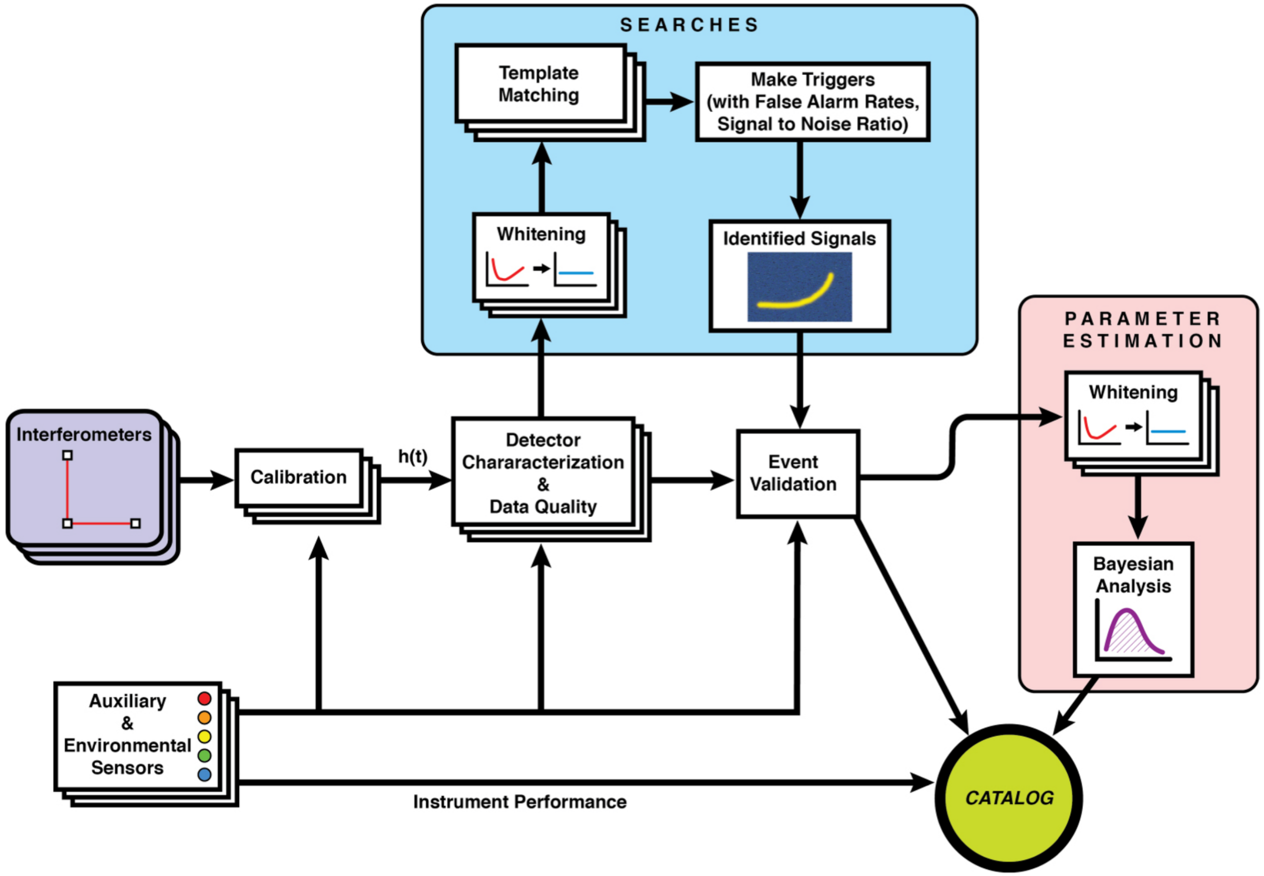
We can then get some useful quantities such as the 90% credible interval

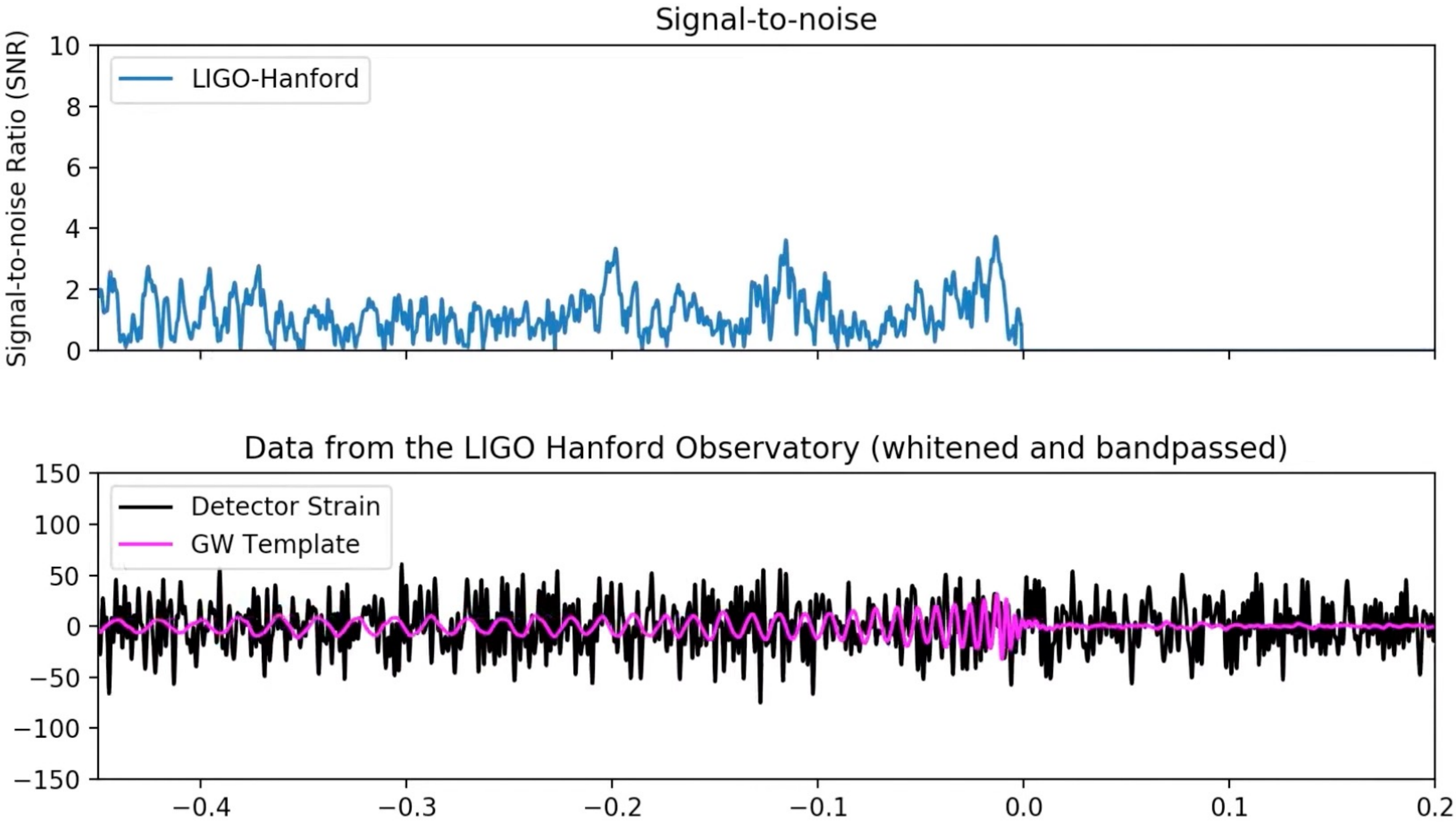
```
lower_bound = np.quantile(Mc, 0.05)
upper_bound = np.quantile(Mc, 0.95)
median = np.quantile(Mc, 0.5)
print("Mc = {} with a 90% C.I = {} -> {}".format(median, lower_bound, upper_bound))
```

[문제4] 주어진 중력파 자료파일에서 bilb

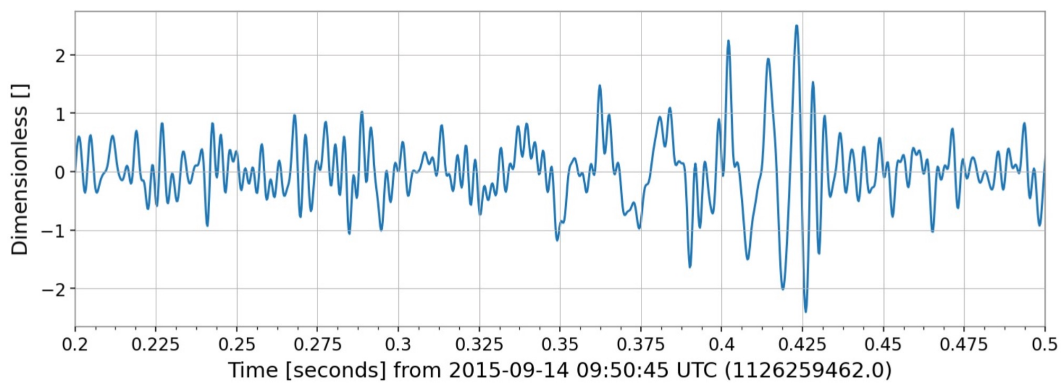
주어진 중력파 자료 파일(H1_GW_NR2024_PR
L1_GW_NR2024_PROBLEM3.gwf)에 채널명 각각 H1:NR2024_WINTER, L1:NR2024_WI
으로 저장된 중력파 신호 파일에서 신호가 갖고 있는 질량 m1, m2, a1z, a2z, 그리고
m2에 대한 표준편차를 계산하여 제출한다. 문제를 풀기 위해서는 https://github.com
[odw/odw-2023](https://github.com)를 참조한다. 특히 bilby를 활용한 모수추정을 설명한 [odw-2023/Tutorials/Day_3/Tuto_3.2_Parameter_estimation_for_compact_object_mergers.i at main · gw-odw/odw-2023 · GitHub](https://github.com) 를 참고한다. 참고 사이트와의 차이는 우리는
어 있는 gwf 파일을 사용하고 psd도 pycbc.psd.AdvDesignSensitivityT1800044 를 사
는 점이다. 그리고 신호의 도착시간은 문제 3에서 찾은 값을 사용한다. 우리가 모수추
값은 m1,m2,a1z, a2z 이지만 실제로 모수 추정을 위한 posterior sample을 생성할 때
리 D까지 같이 해야 한다. 생성된 posterior sample을 이용하여 m1, m2에 대한 corne
을 그리면 m1, m2의 값과 표준 편차를 알 수 있다. 상한과 하한이 다를 경우에는 하한
제출한다. a1z, a2z의 값은 a_1, a_2 에 대한 corner plot을 그려서 그 값을 알 수 있다.
목 값마다 5점씩 부여한다. 표준편차의 값은 실제로 90% confidence interval을 계산한 후에
중간 값에서 하한 값을 뺀 값을 제출한다. 계산한 posterior sampler에서 중간값(median)과
90% confidence interval을 계산하는 예제는 위에 언급한 사이트를 참고하시오.

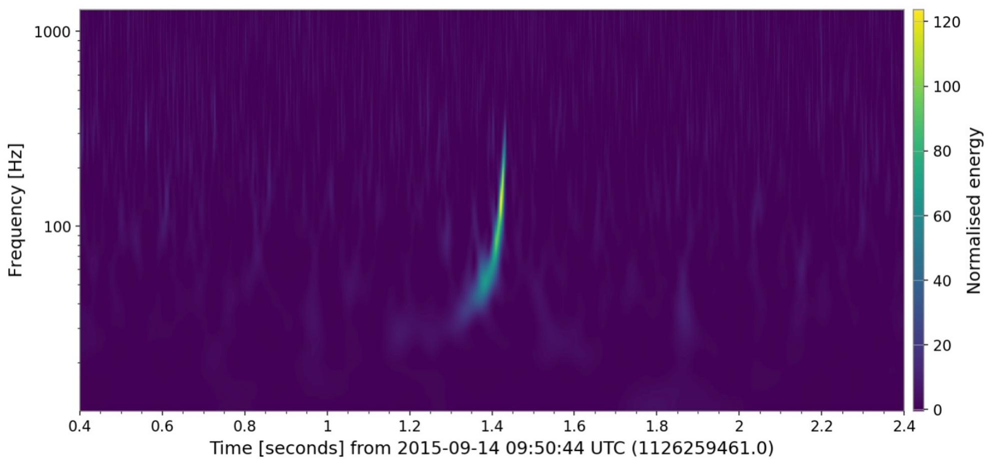



SEARCHES
Template Matching
Make Triggers (with False Alarm Rates, Signal to Noise Ratio)
Whitening
Identified Signals
PARAMETER ESTIMATION
Interferometers
Calibration
h(t)
Detector Chararacterization & Data Quality
Event Validation
Whitening
Bayesian Analysis
Auxiliary & Environmental Sensors
Instrument Performance
CATALOG



Signal-to-noise
Signal-to-noise Ratio (SNR)
LIGO-Hanford
10
8
6
4
2
0
Data from the LIGO Hanford Observatory (whitened and bandpassed)
Detector Strain
GW Template
150
100
50
0
−50
−100
−150
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2


Dimensionless []
Time [seconds] from 2015-09-14 09:50:45 UTC (1126259462.0)


Frequency [Hz]
Normalised energy
Time [seconds] from 2015-09-14 09:50:44 UTC (1126259461.0)