

# The Parameter Estimation in PyCBC Inference

2019/05/10

Guo-Zhang Huang(NTNU)

# Outline

- Bayesian Inference
  - Bayesian
  - Likelihood
  - MCMC
- PyCBC Inference workflow
- The simple example for normal distribution
- Basic background for *pycbc\_inference*
- GW150914 Inference

# Bayesian Inference

Bayesian methods are used to infer the properties of detected astrophysical sources

$$p(\vec{\vartheta}|\vec{d}(t), H) = \frac{\overset{\text{likelihood}}{p(\vec{d}(t)|\vec{\vartheta}, H)}p(\vec{\vartheta}|H)}{p(\vec{d}(t)|H)} \quad \text{posterior probability density}$$

$\vec{d}(t)$  the observed data

$H$  is the model of the gravitational-wave signal

$\vec{\vartheta}$  are the parameters of this model

# Likelihood function

The form of  $p(\vec{d}(t)|\vec{\vartheta}, H)$  is well-known likelihood for a signal of known morphology in Gaussian noise.

$$p(\vec{d}(t)|\vec{\vartheta}, H) = \exp\left[-\frac{1}{2} \sum_{i=1}^N \langle \tilde{n}_i(f) | \tilde{n}_i(f) \rangle\right]$$

The equation use matched-filter to calculate signal-to-noise ratio in PyCBC Inference.

# Markov chain Monte Carlo (MCMC)

It have been used to numerically sample the posterior probability density function of astrophysical parameters for data.

The true astrophysical parameters,  $\vec{\theta}$ , can then be estimated from histograms of the position of the Markov chains in the parameter space.

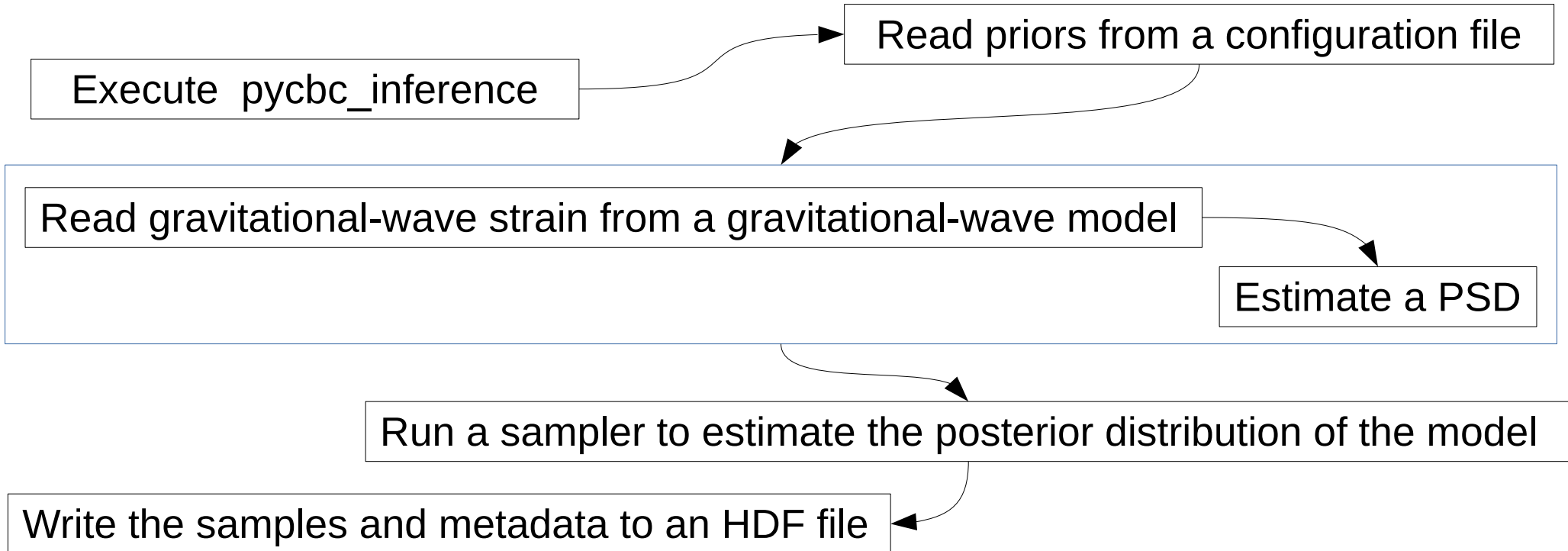
1. An initial Markov chain draws a set of parameter, from the prior probability density function.
2. The sampling algorithm randomly chooses a new set of parameters, and compute with the origin one.
3. If the Markov chain moves to the proposed parameter set, it updates this parameter set.
4. After a sufficient number of iterations, the ensemble converges to a distribution that is proportional to a sampling of the posterior probability density function.

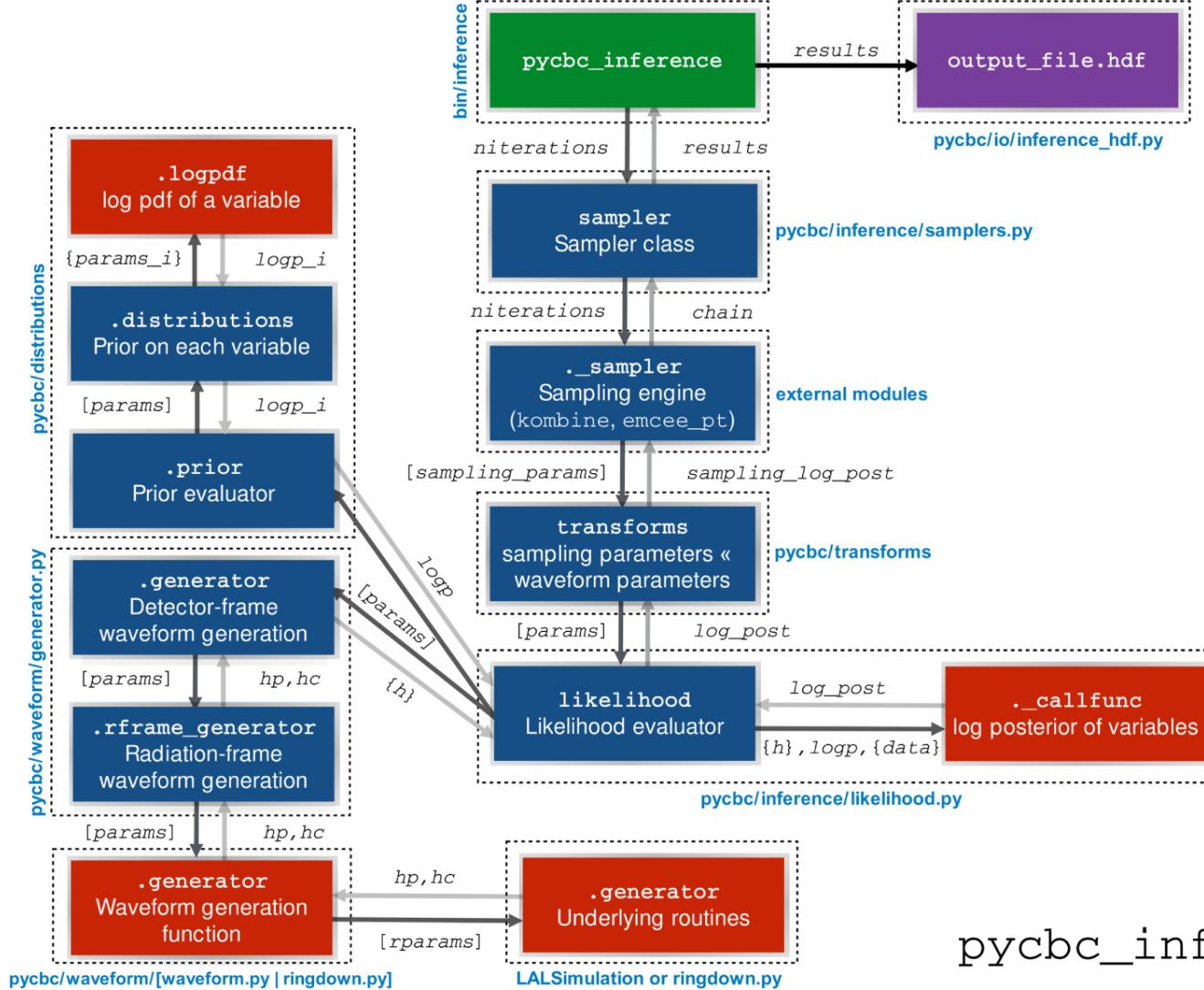
<https://arxiv.org/abs/1807.10312>

# PyCBC Inference

<http://pycbc.org/pycbc/latest/html/>

The executable **pycbc\_inference** is designed to sample the parameter space and save the samples in an HDF file.





**Key**

- █ : Executable
- █ : Class
- █ : Function
- █ : File
- : Module

`iterations`: Number of samples to return  
`[sampling_params]`: List of sampling-frame parameter values to test  
`[params]`: List of parameter values to test  
`[rparams]`: Radiation-frame parameter values  
`{params_i}`: Dictionary of parameters for  $i$ -th prior  
`logp_i`: Log pdf of the  $i$ -th prior at `{params_i}`  
`logp`: Log of the prior probability at `[params]`  
`hp, hc`: Plus and cross polarization of the waveform at `[rparams]`  
`{h}`: Dictionary of the waveform at `[params]` in each detector  
`{data}`: Dictionary of the data in each detector  
`log_post`: Log of the posterior probability at `[params]`  
`sampling_log_post`: Log of the posterior probability in the sampling-frame  
`chain`: Accepted parameters returned by the sampling engine.  
 FieldArray of accepted `[params]` over `iterations` and their `log_post` values

pycbc\_inference Call Graph



# The simple example for normal distribution

normal2d.ini

```
[model]
name = test_normal

[sampler]
name = emcee
nwalkers = 5000
niterations = 100

[variable_params]
x =
y =

[prior-x]
name = uniform
min-x = -10
max-x = 10

[prior-y]
name = uniform
min-y = -10
max-y = 10
```

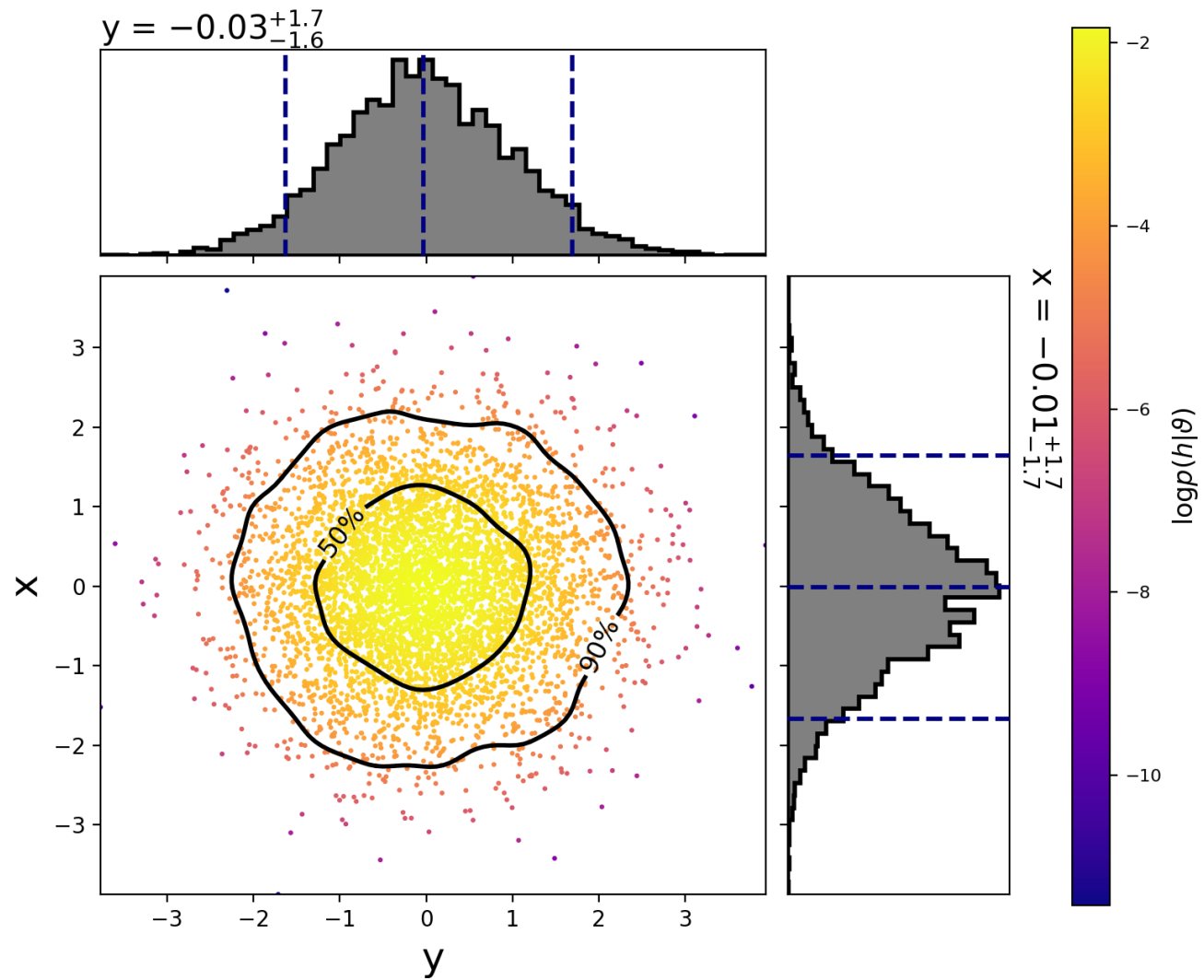
This simple example shows how to inference the 2D normal distribution by `pycbc_inference`.

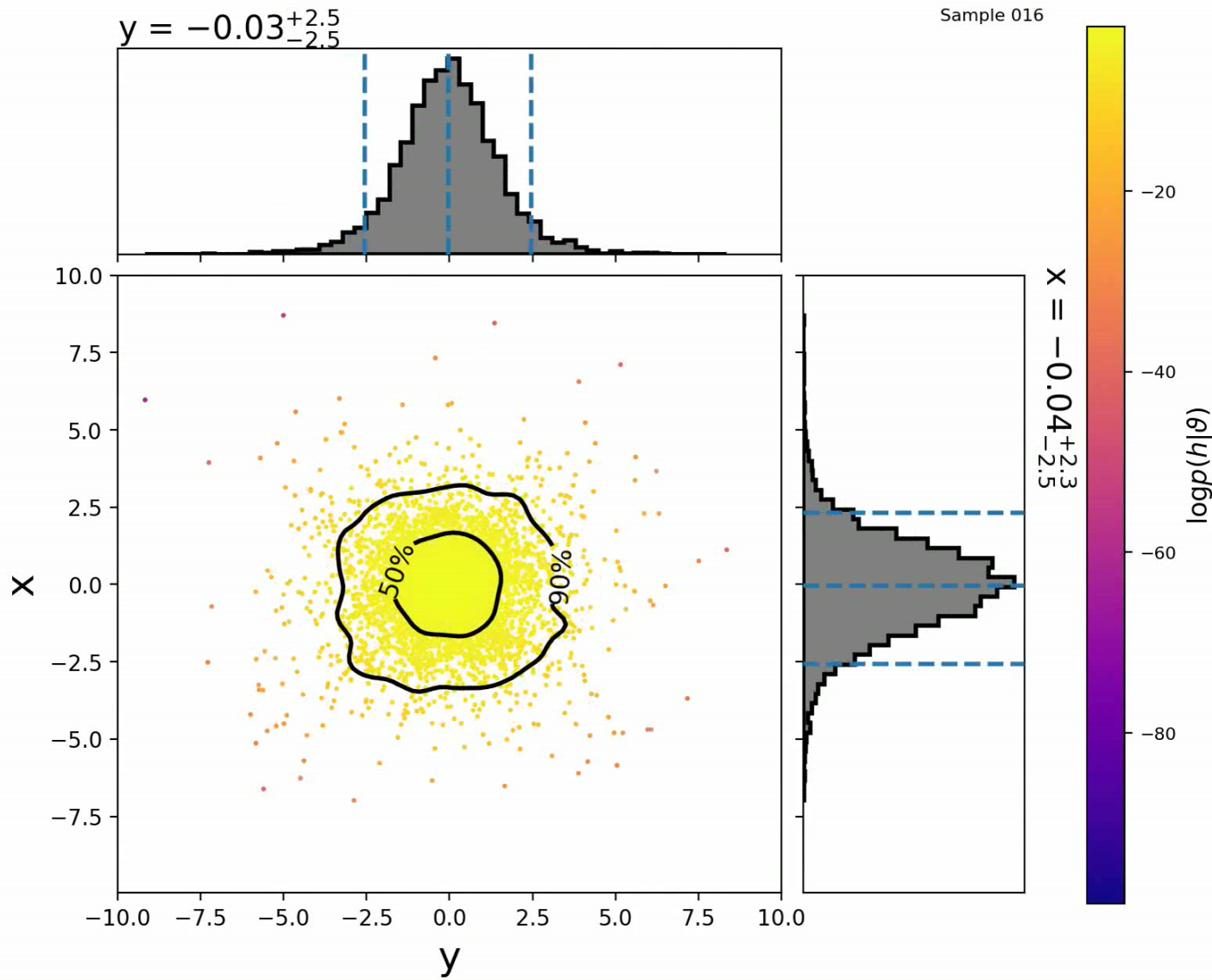
First, the priors of probability are the normal distribution.

Second, the sampler uses the `[emcee]` to sampling points on the parameter space.

The `emcee` is a python package, and it is the MCMC algorithm.

The 5000 walkers sampling on the parameter space for 100 times.





# Configuration File

The configuration file contains information about Inference in the whole workflow.

The main section of the configuration file is [model], [sampler] and [prior].

# Configuring model

The [model] section sets up what model to use for the analysis.

```
[model]
name = gaussian_noise
```

The BBH case usually uses the Gaussian noise model to do parameter estimation. The PyCBC Inference assumes that each detector produces stationary Gaussian noise, that is uncorrelated between the detectors.

# Configuring sampler

The [sampler] section sets up what sampler to use for the analysis.

The sampler defines that walkers sampling points and iterations on whole parameter space.

The toolkit has two algorithms, one is the [emcee], the other is [emcee\_pt].

```
[sampler]
name = emcee
nwalkers = 5000
niterations = 1000
checkpoint-interval = 100
```

This case consider the 5000 walkers for 1000 iterations, and it checkpoint for every 100th iterations.

# Configuring prior

The prior parameter is determined by the waveform model.

The mass parameters are an important variable in the gravitational waveform, so the mass is the prior parameter for posterior distribution.

```
[variable_params]
```

```
x =  
y =
```

The sample a posterior over two parameters called x and y.

```
[prior-x]
```

```
name = uniform  
min-x = -10  
max-x = 10
```

```
[prior-y]
```

```
name = uniform  
min-y = -10  
max-y = 10
```

The name represent to different distribution.

A uniform distribution on the given parameters.

The parameters are independent of each other.

The section of [static params] is input the known parameter on initial setting.

```
[static_params]
approximant = IMRPhenomPv2
f_lower = 18
```

The setting of priors is constrained by the waveform model.

Some waveform model set variable parameters in the special range, like mass ratio  $< 2$ .

If you fix some known parameter for sampling, the process of PE will reduce costing of time.



# Sampling transforms

One or more of the [variable\_params] may be transformed to a different parameter space for purposes of sampling.

```
[sampling_params]
mass1, mass2: mchirp, q

[sampling_transforms-mchirp+q]
name = mass1_mass2_to_mchirp_q
```

You can search sampling transforms on the PyCBC homepage, and you see the available list on there.

# GW150914 example

Thanks for your attention