PHYS 139 Final Project

---- Gravitational Waves Detection

Group A: William, Nani, Leo, Yuntong

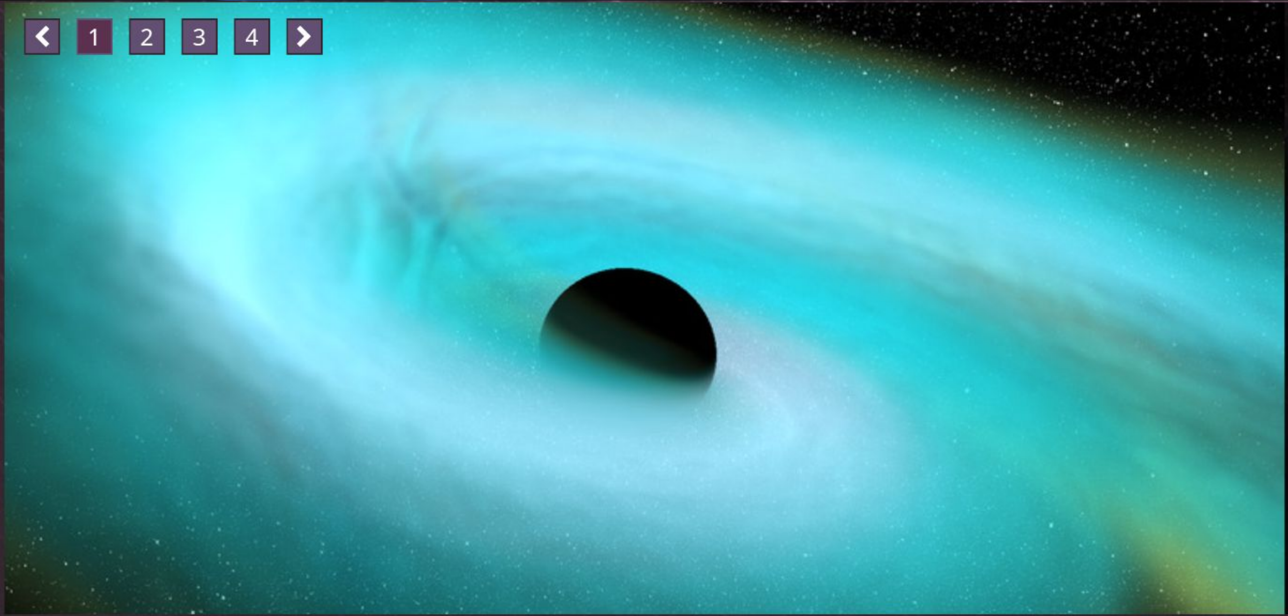
Catching gravitational waves

- First observed on 14 September 2015 by LIGO -- GW150914
- Produced by the merger of a pair of blackholes
- Ripples in spacetime
- 2 other detections, 15 June 2016
- 8 detections, 2017

Implications:

- Probe into the conditions of the earliest universe
- Study the nature of dark energy

Recent Events



LIGO-Virgo-KAGRA Finds Elusive Mergers of Black Holes with Neutron Stars

News Release • June 29, 2021

LVK scientists capture two NSBH merger events 10 days apart in January 2020.

Implications of lambda parameters

Gravitational-wave detection rates for compact binaries formed in isolation: LIGO/Virgo O3 and beyond

Vishal Baibhav, Emanuele Berti, Davide Gerosa, Michela Mapelli, Nicola Giacobbo, Yann Bouffanais, Ugo N. Di Carlo

Using simulations performed with the population synthesis code MOBSE, we compute the merger rate densities and detection rates of compact binary mergers formed in isolation for second- and third-generation gravitational-wave detectors. We estimate how rates are affected by uncertainties on key stellar-physics parameters, namely common envelope evolution and natal kicks. We estimate how future upgrades will increase the size of the available catalog of merger events, and we discuss features of the merger rate density that will become accessible with third-generation detectors.

Measuring the Star Formation Rate with Gravitational Waves from Binary Black Holes

Salvatore Vitale[1,2], Will M. Farr[3,4,5], Ken K. Y. Ng[1,2], and Carl L. Rodriguez[2,6]

Published 2019 November 12 •

The Astrophysical Journal Letters, Volume 886, Number 1

Citation Salvatore Vitale *et al* 2019 *ApJL* **886** L1

DOI 10.3847/2041-8213/ab50c0

Gibbs Sampling

- a Monte Carlo Markov Chain method that iteratively draws an instance from the distribution of each variable, conditional on the current values of the other variables in order to estimate complex joint distributions.

Technical Report | Published: 03 June 2021

Rapid genotype imputation from sequence with reference panels

Inexpensive genotyping methods are essential to modern genomics. Here we present QUILT, which performs diploid genotype imputation using low-coverage whole-genome sequence data. QUILT employs Gibbs sampling to partition reads into maternal and paternal sets, facilitating rapid haploid imputation using large reference panels. We show this partitioning to be accurate over many megabases, enabling highly accurate imputation close to

Monte Carlo Gibbs Sampling

- Ultimate goal is to find the predicted time of model change n_o:
- For an initially chosen value n_o
- Iterate through finding the maximum probability values of λ_1, λ_2 given n_o
- Use those values to find the new maximum probability value of n_o given λ_1, λ_2
- Repeat y ~ 1000 times to get an accurate prediction for a single point
- Iterate over N points to get a histogram of the probability distribution for n_o, λ_1, and λ_2

Having obtained $\boldsymbol{x}_{1:N} := \{x_1, \dots, x_N\}$, select a and b.
Initialize $n_0^{(0)}$
For $i = 1, 2, \dots,$ **Do**
- $\lambda_1^{(i)} \sim \text{Gamma}(\lambda | a + \sum_{n=1}^{n_0^{(i-1)}} x_n,\ b + n_0^{(i-1)})$
- $\lambda_2^{(i)} \sim \text{Gamma}(\lambda | a + \sum_{n=n_0^{(i-1)}+1}^{N} x_n,\ b + (N - n_0^{(i-1)}))$
- $n_0^{(i)} \sim P(n_0 | \lambda_1^{(i)}, \lambda_2^{(i)}, x_{1:N})$

End For

Implementation lambda_1 and lambda_2

```
def lambdacalc(data, n0, a, b, is1):

   if(is1):

      # use data from 0 to n0

      a1 = a + np.sum(data[:(n0+1)])

      b1 = b + n0

      return np.random.gamma(a1,1/b1), a1, b1

   else:

      # use data from n0 + 1 to N

      a2 = a + np.sum(data[(n0+1):])

      b2 = b + (data.size -  n0)

      return np.random.gamma(a2,1/b2), a2, b2
```

$$p(\lambda_1|n_0, \lambda_2, \boldsymbol{x}_{1:N}) = \mathrm{Gamma}(\lambda_1|a_1, b_1),$$

$$a_1 = a + \sum_{n=1}^{n_0} x_n, \quad b_1 = b + n_0,$$

$$p(\lambda_2|n_0, \lambda_1, \boldsymbol{x}_{1:N}) = \mathrm{Gamma}(\lambda_2|a_2, b_2),$$

$$a_2 = a + \sum_{n=n_0+1}^{N} x_n, \quad b_2 = b + (N - n_0),$$

Implementation n0

a) create List holding the 184 discrete probabilities

b) calculate the normalization as we derived in part a)

c) calculate each discrete probability corresponding to the years, and store the value in a list

d) apply the random.choices() function that sample new n0 from the discrete probability

e) return the new n0 value

$$\ln P(n_0|\lambda_1,\lambda_2,x_{1:N}) \cong \ln\lambda_1 \sum_{n=1}^{n_0} x_n - n_0\lambda_1 + \ln\lambda_2 \sum_{n=n_0+1}^{N} x_n - (N-n_0)\lambda_2, \quad n_0 = 1,2,\ldots,N.$$

Details on the proof in our github:

https://github.com/JoTimelord/PHYS_139_Final_Team_Project/blob/main/PHYS_139_Final_Project_Analysis_Note_Group.pdf

Result

Initial values:

Lambda_1: 8

Lambda_2: 1

N0: 2022

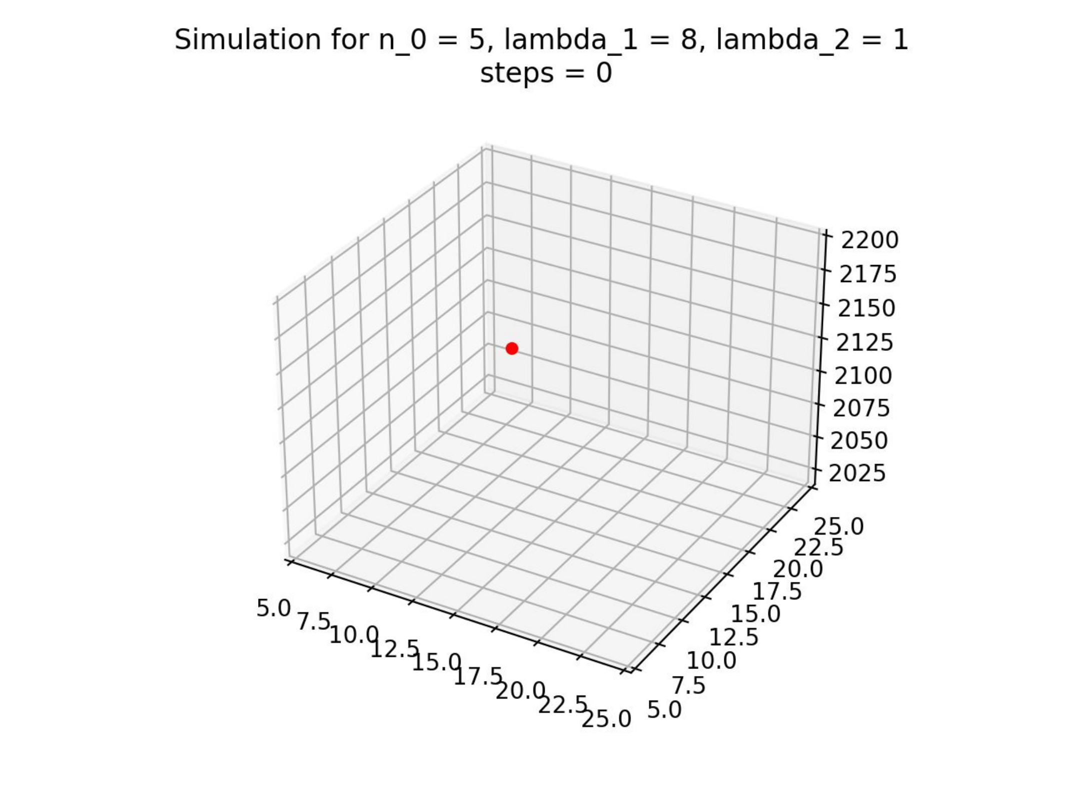
Iterations: 1000 times

Means:

Lambda_1 = 10.514275524638396

Lambda_2 = 18.06408117987173

N0 = 2109.585



Result: n0 probability distribution

Initial values:

Lambda_1: 8

N0: 2022

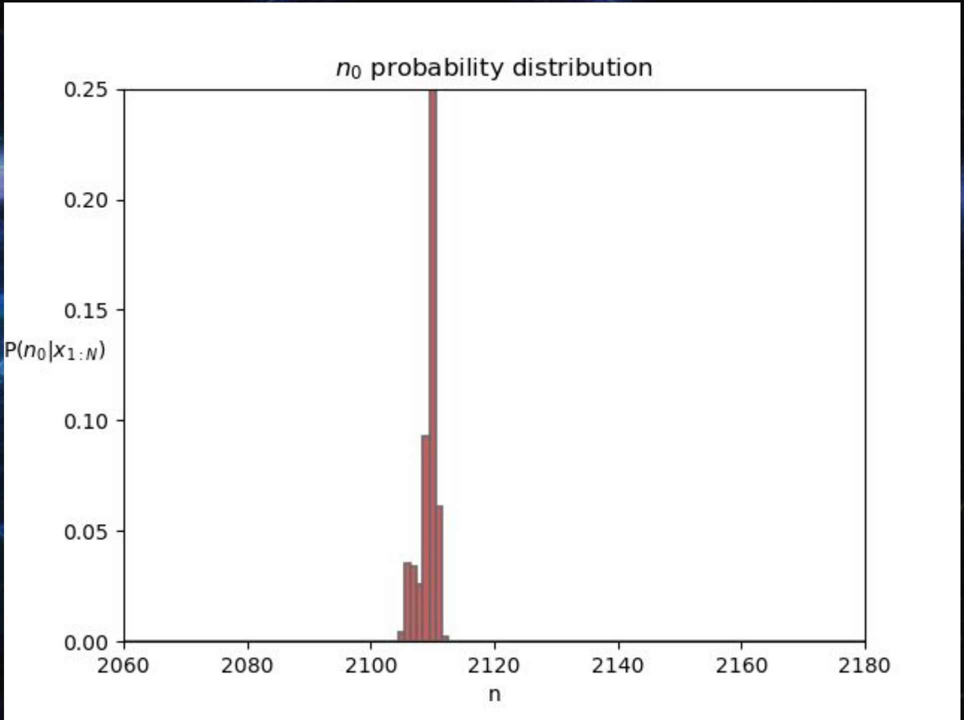
Iterations: 5000 times

Means:

Lambda_1 = 10.517138806777954

Lambda_2 = 18.113973557722252

N0 = 2109.668



Result: lambda_1 and lambda_2 from Gibbs sampling

Initial values:

Lambda_1: 8

N0: 2022

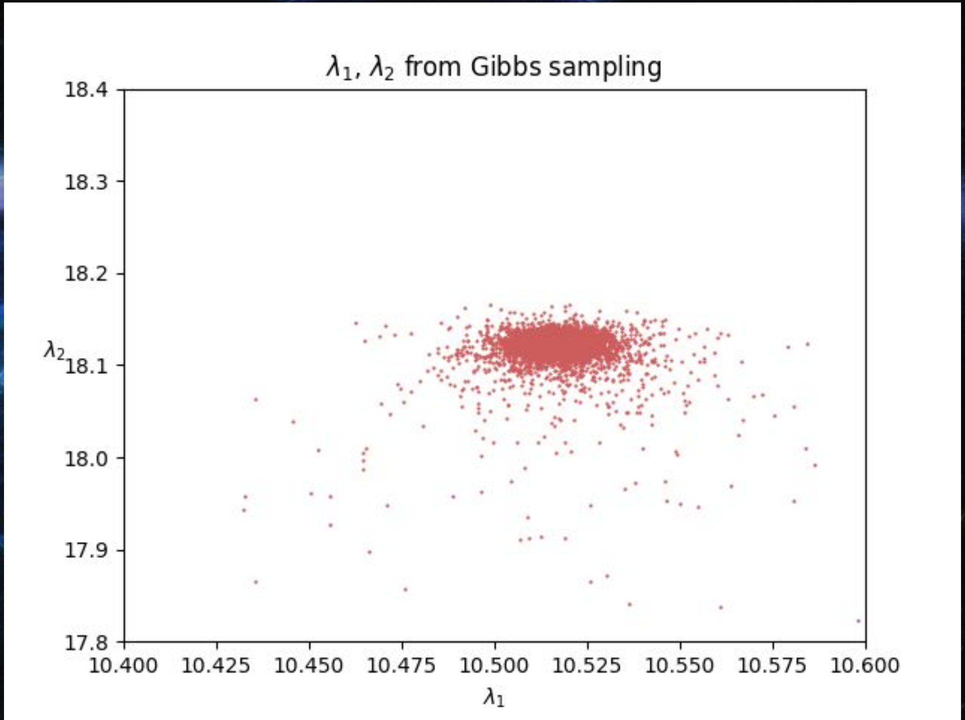
Iterations: 5000 times

Means:

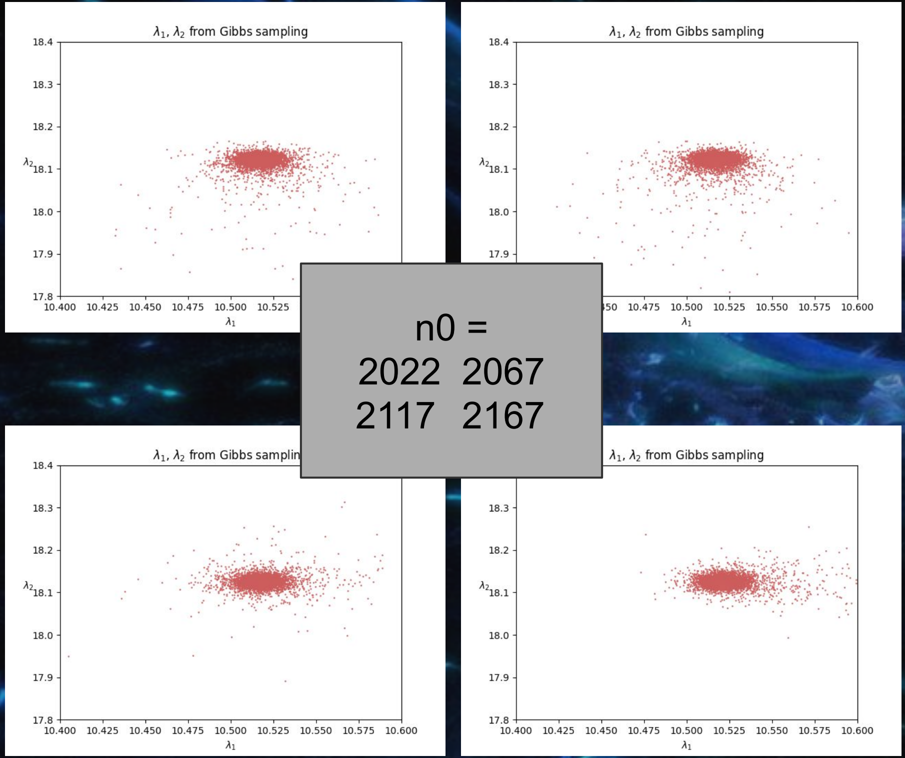
Lambda_1 = 10.517138806777954

Lambda_2 = 18.11397355772252

N0 = 2109.668



Result



Initial values:

Lambda_1: 8

N0: 2022

Iterations: 5000 times

Means:

Lambda_1 = 10.517138806777954

Lambda_2 = 18.11397355772252

N0 = 2109.668

Result

Initial values:

Lambda_1: 8

n0: 2022-2167

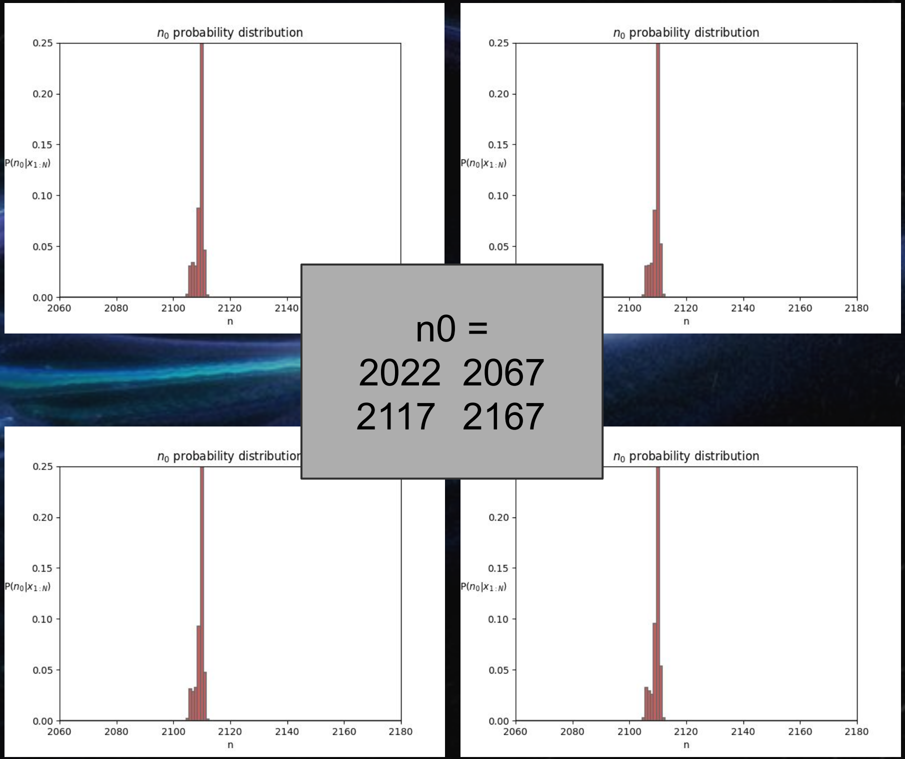
Iterations: 5000 times

Means:

Lambda_1 = 10.517138806777954

Lambda_2 = 18.113973557722252

N0 = 2109.668



n0 =
2022 2067
2117 2167