Effects of Different Data Quality Veto Methods in the PyCBC Search for Compact Binary Coalescences

Brina Martinez
University of Texas Rio Grande Valley
Mentor: Dr. Derek Davis
LIGO Laboratory, California Institute of Technology

LIGO SURF 2020
August 21, 2020

Overview

- Goals of the project
- PyCBC search pipeline
- Data quality vetoes
- Current veto methods
- New methods
- Results
- Potential extension for project

Goals

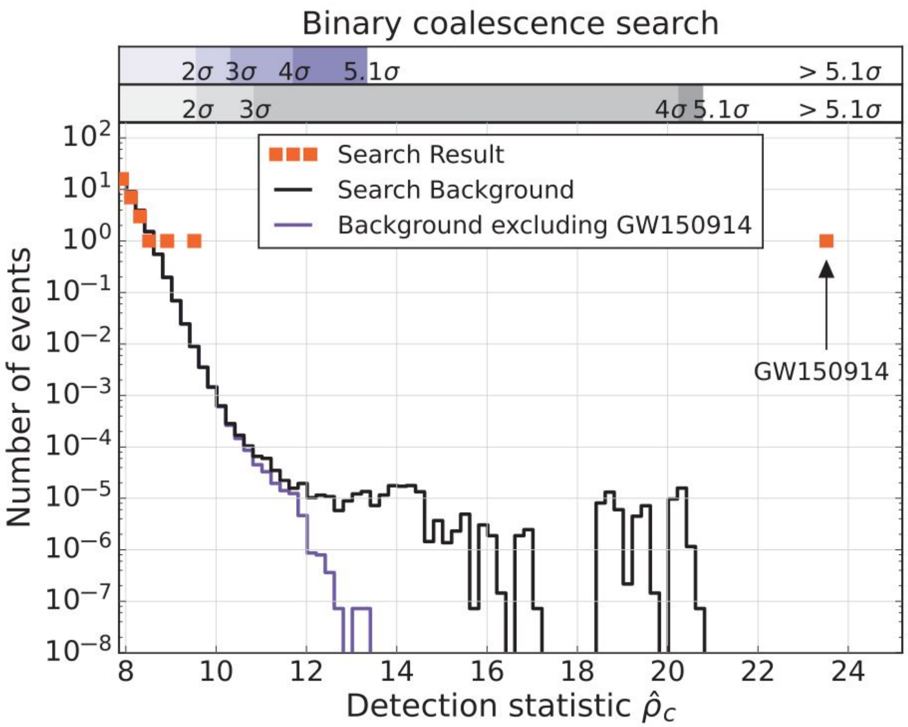
We want to confidently mitigate noise in the detector's data and finely tune our pipeline to prevent a decrease in search sensitivity and detect more signals!

- What can cause a decrease in search sensitivity?
  - Loud glitches
  - Removing too much data (time)
  - Using ineffective flags

PyCBC search pipeline

- The PyCBC pipeline is used in the search for Gravitational Waves from a Compact Binary Coalescence
  - Compact Binary Coalescences are when two compact objects such as black holes or neutron stars coalesce and experience an inspiral, merger, and ringdown.
- PyCBC shows us the signal-to-noise ratio and ranking statistics for triggers and signals in data when correlated with expected waveforms
- PyCBC uses matched filtering to match signals to gravitational wave templates
- PyCBC uses a chi-squared consistency test to downrank triggers in the data that are due to glitches and increase the significance of real GW signals. The SNRs of triggers are weighted along this consistency test and outputs a new ranking statistic or re-weighted SNR

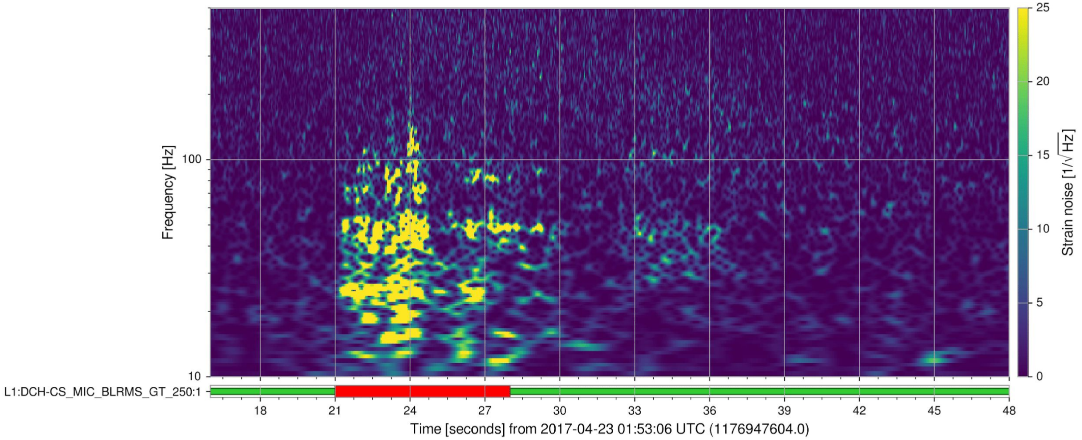
PyCBC search pipeline



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

Data quality vetoes

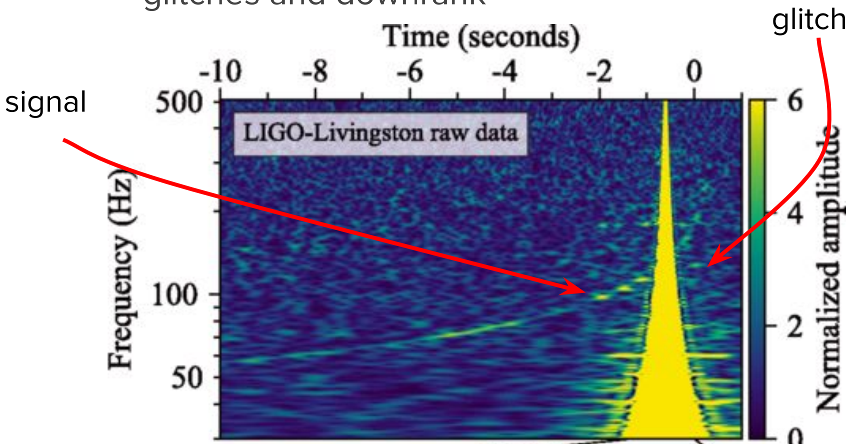
- Vetoes are generated from flags which identify transient noise in data where glitches or noise make it difficult to run analysis on signals.
- In order for a flag to identify these transient noises they need to be correlated with some disturbance in or around the detector
    - These are referred to as “CAT2 Flags”
- Vetoes remove flagged segments to improve the analysis pipeline’s search for signals



Current Method vs Improved Method

Current Method

- Removes glitches and flagged times completely
- If flags are not as efficient, they do not highlight enough glitches, decreasing search sensitivity
- Uses chi-square consistency test to analyze glitches and downrank



B.P. Abbott *et al.* Phys. Rev. Lett. 119, 161101

Improved Method

- Our method shows an effective glitch veto that increases the significance of signals and the overall number of detectable signals without removing data.
- Keeps glitches that are flagged, removing no data
- Uses chi-square consistency test and re-ranking of the glitch statistic

How is this done?

- Uses CAT2 data quality vetoes
- Uses Likelihood ratio of trigger rate for vetoed time vs all time to re-rank data against original background

Likelihood in the Improved Method

- How much more likely is a trigger to show up during a flag vs all time?

$$\mathcal{L}(flag) = \frac{\mathcal{L}(flagtime)}{\mathcal{L}(totaltime)}$$

$$\frac{\mathcal{L}(t|\theta) = \dfrac{triggers\ flagged}{flagged\ duration} * \dfrac{1}{triggers\ total}}{\mathcal{L}(t|\theta) = \dfrac{triggers\ total}{total\ duration} * \dfrac{1}{triggers\ total}}$$

```
Number of triggers: 5243
Number of flagged times: 637
Total known time: 714761
Total active time of flags: 3338.0
Likelihood of total time: 0.000001399
Likelihood of flags: 3.6397641432835155e-05
Likelihood ratio: 26.017155111
```

Ranking in Improved Method

Re-ranking triggers

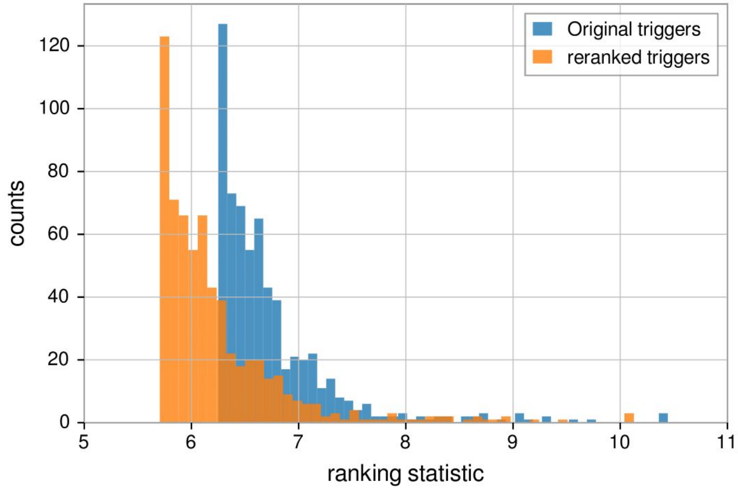
$$\widetilde{\rho} = \sqrt{\rho^2 - 2ln\mathcal{L}}$$

New ranking statistic

Original ranking statistic

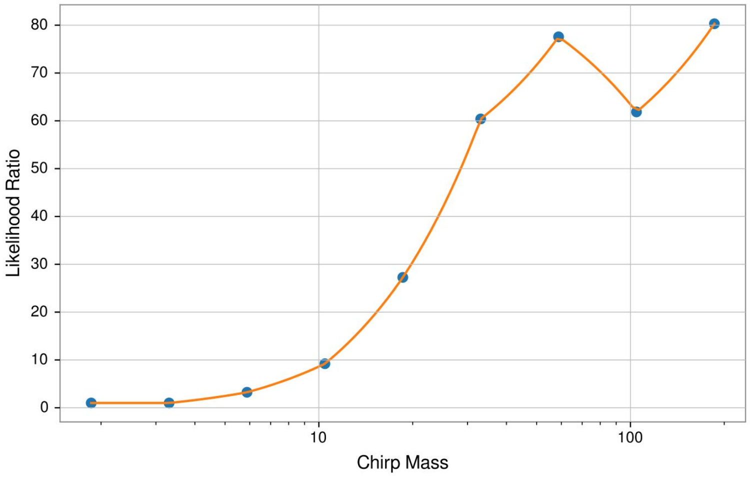
Likelihood ratio of data quality flag

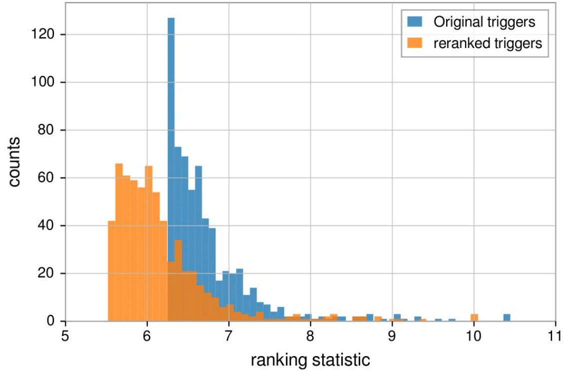
- Ranking statistic of triggers during vetoed times are updated using the likelihood ratio

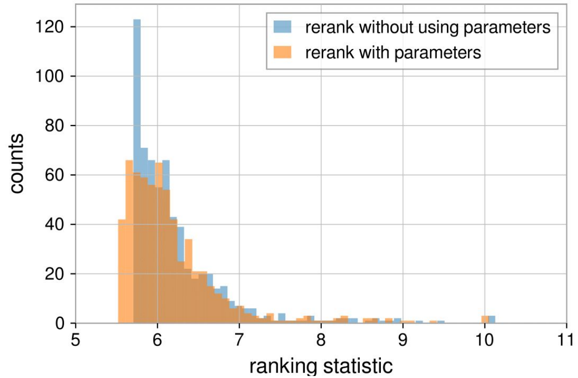


Using different parameters to calculate likelihood

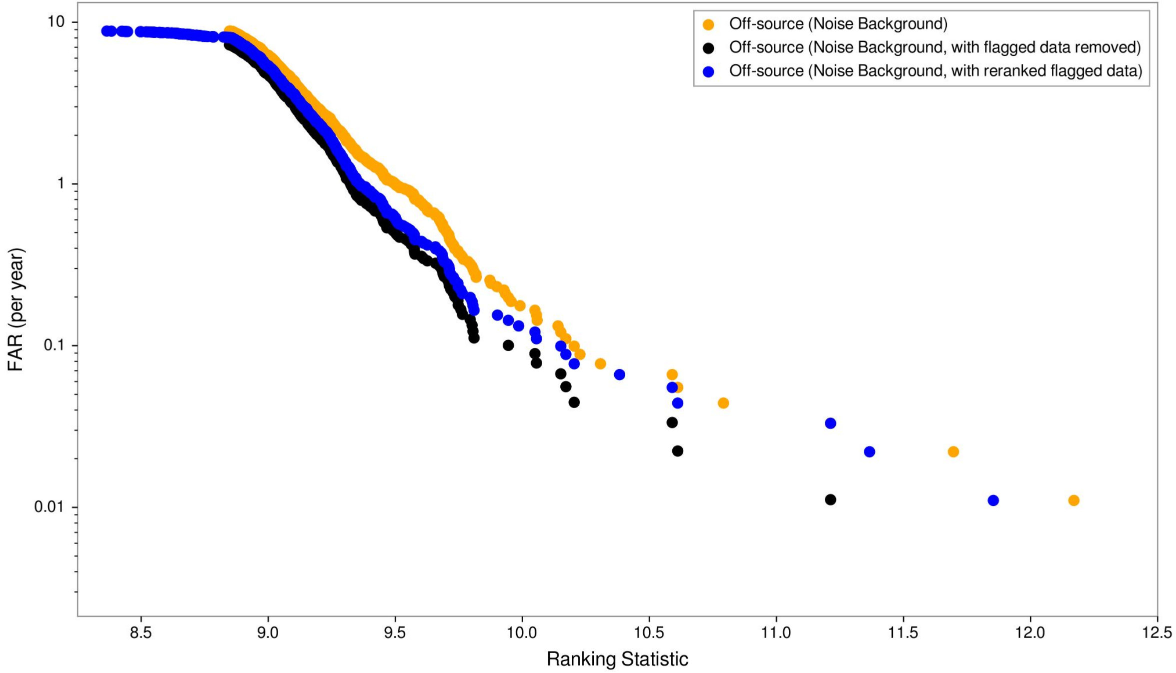
- **We calculated the likelihood ratio for different chirp masses**
    - Other template parameters we could use
        - Mass ratio
        - Template duration







Results




```
original vs flagged comparison
The ratio of distance:          1.02
The ratio of time:              0.99
The ratio of volume * time:     1.04
original vs reranked comparison
The ratio of distance:          1.01
The ratio of time:              1.00
The ratio of volume * time:     1.04
flagged vs reranked comparison
The ratio of distance:          1.00
The ratio of time:              1.01
The ratio of volume * time:     1.00
```

Conclusions

Improved veto method vs. Current veto method:

- Re-ranks flagged data instead of removing flagged data completely
- Increases the significance of signals
- Reduces the time lost due to flags
- **Increases the overall number of detectable signals**

LIGO NSF

Future Investigations:

- Expand amount of flags applied
- Expand to different DQ products (e.g. iDQ, Gravity Spy)
- Expand to updated PyCBC ranking statistic

Thank you! Questions?

I would like to give a special thank you to Derek for being an amazing mentor, Dr. Weinstein for endless support and enthusiasm, the National Science Foundation, the Caltech LIGO SURF Program, and LIGO Laboratory.

Computing support for this project was provided by the LDAS computing cluster at the California Institute of Technology. LIGO was constructed by the California Institute of Technology and Massachusetts Institute of Technology with funding from the National Science Foundation, and operates under cooperative agreement PHY-0757058. The LIGO SURF Program is supported by NSF award PHY-1852081.

References

This project uses PyCBC to investigate impacts of different veto methods on search sensitivity

- B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton. FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries. Phys. Rev. D, 85:122006, 2012.
- Bruce Allen. A chi**2 time-frequency discriminator for gravitational wave detection.Phys.Rev. D, 71:062001, 2005.
- Alexander H. Nitz, Thomas Dent, Tito Dal Canton, Stephen Fairhurst, and Duncan A. Brown. Detecting binary compact-object mergers with gravitational waves: Understanding and Improving the sensitivity of the PyCBC search. Astrophys. J., 849(2):118, 2017.
- Tito Dal Canton et al. Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors. Phys. Rev. D, 90(8):082004, 2014.
- Samantha A. Usman et al. The PyCBC search for gravitational waves from compact binary coalescence. Class. Quant. Grav. , 33(21):215004, 2016.
- Alexander H. Nitz, Tito Dal Canton, Derek Davis, and Steven Reyes. Rapid detection of gravitational waves from compact binary mergers with PyCBC Live. Phys. Rev. D, 98(2):024050, 2018

Additional sources

- B P Abbott et al. Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO's first observing run. Class. Quant. Grav., 35(6):065010, 2018.
- Godwin et al. Incorporation of Statistical Data Quality Information into the GstLAL Search Analysis. LIGO Document number P2000117
- B P Abbott *et al* Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914. 2016 *Class. Quantum Grav.* 33 134001

Extra slides

LIGO

Math!

$$\rho^2 \propto 2\left[\log p^S(\theta) - \log p^N(\theta)\right]$$

Ranking statistic

Signal distribution

Noise distribution

$$p^N(\theta) \propto \lambda^N_H(\hat{\rho}_L, \tau)\lambda^N_H(\hat{\rho}_L, \tau)$$

Single-IFO noise model

Single-IFO ranking statistic

Template duration

$$\lambda^N_{Aw}(\hat{\rho}_A) = \mu_{Aw}\alpha_{Aw}\exp[-\alpha_{Aw}(\hat{\rho}_A - \hat{\rho}_{th})]$$

Total noise rate

Distribution steepness

Threshold ranking statistic

$$\mu^N_{Aw} \rightarrow \mu^N_{Aw}(\vec{\theta}_{\mathrm{DQ}})$$

Equations from O2 stat paper:
Nitz et al. Astrophys.J. 849 (2017) no.2, 118

DQ timeseries

$$\vec{\theta}_{DQ} = \text{iDQ}$$

Input data is iDQ timeseries

Lambda calculated using iDQ likelihood vs cumulative trigger counts

Similar implementation to Godwin et al. (P2000117)

$$\vec{\theta}_{DQ} = \text{iDQ streams + DQ flag(s)}$$

Input data is multiple semi-independent data streams

Lambda calculated for each input data stream independently

“Final” lambda is maximized for each individual trigger