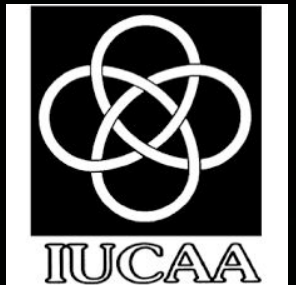
PyCBC

HSF INDIA – IUCAA JOINT WORKSHOP

17th April 2026

Rahul Dhurkunde, Max Trevor

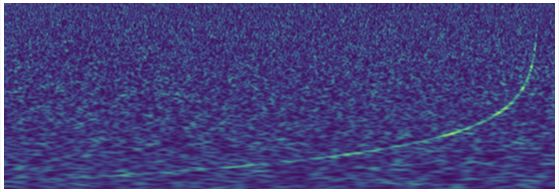
University of Portsmouth
University of Maryland



PyCBC - Lecture 1

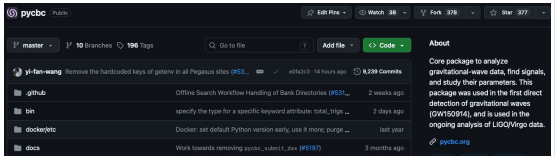
Max Trevor, Rahul Dhurkunde

Overview & Impact



Scientific Legacy

Used in the first direct detection of Gravitational Waves (GWs).



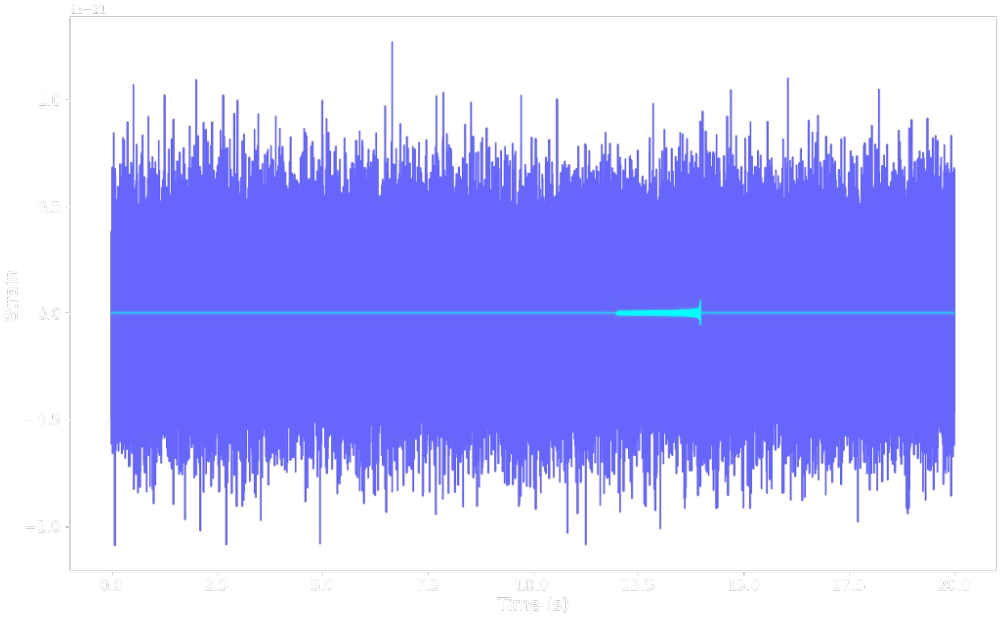
Open Ecosystem

A core package to analyze data, find signals, and study parameters. Fully open-source software.

Community

Largest number of active users and contributors amongst any GW software.

Challenges of GW detection (in a nutshell)

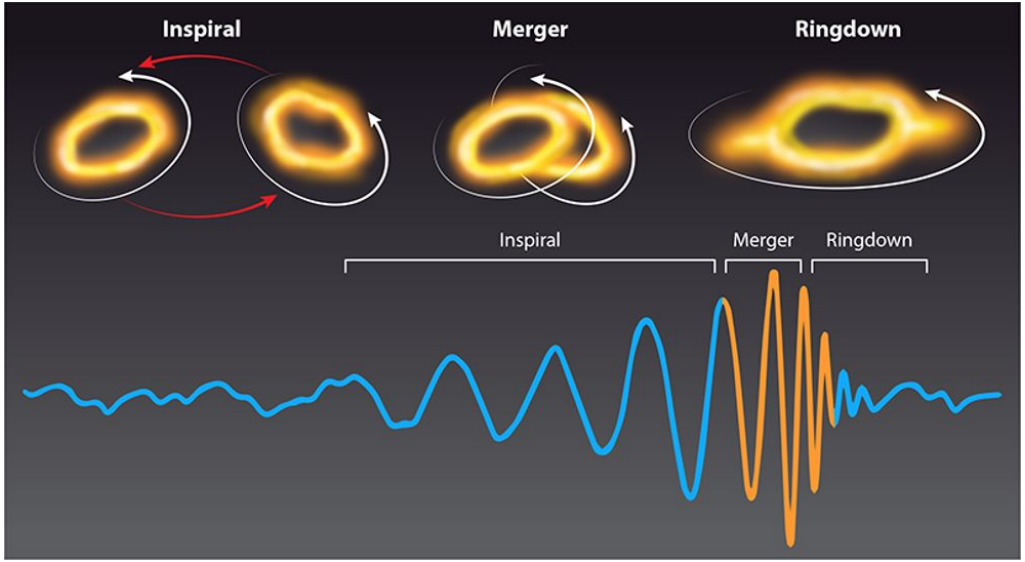


Data	Signal
~ Years	~ seconds
~ Terabytes of data	~ 10 Kb of a signal

Fourier domain

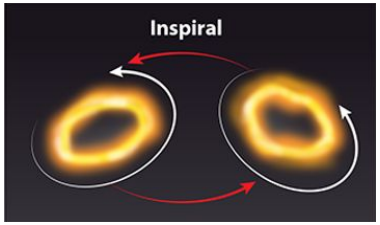
CHARLES PLATIAU / Reuters

Signal modelling

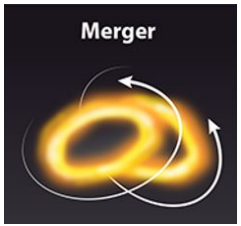


(Top) Kip Thorne; (Bottom) B. P. Abbott et al. [8]; adapted by APS/Carin Cain

Signal modelling

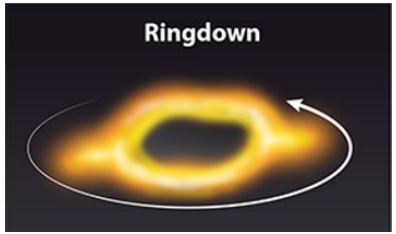


Post-Newtonian approximation



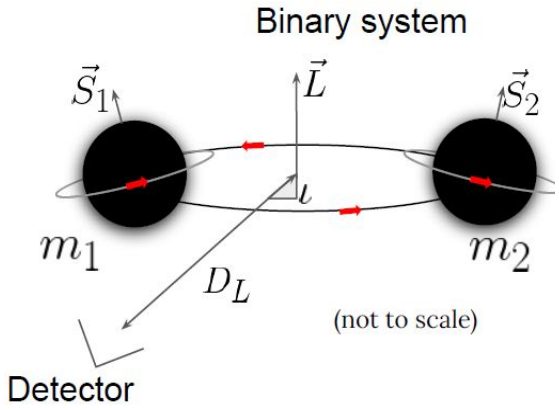
$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

Numerical relativity



Damped sinusoids

Modelling compact binaries



Component masses

Component spin vectors

Sky location

Luminosity distance

Inclination of the orbit

Orbital phase

Polarization angle

Signal's time of arrival

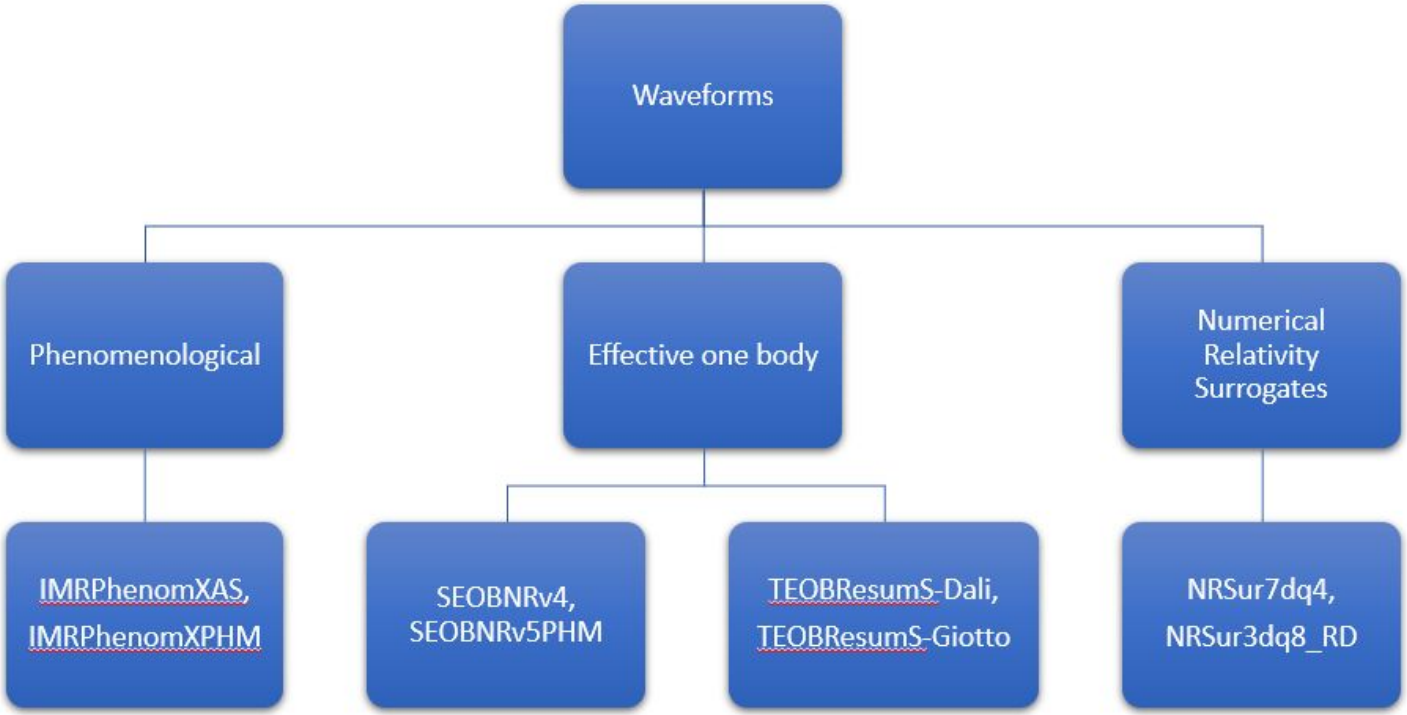
Orbital eccentricity

Tidal deformability

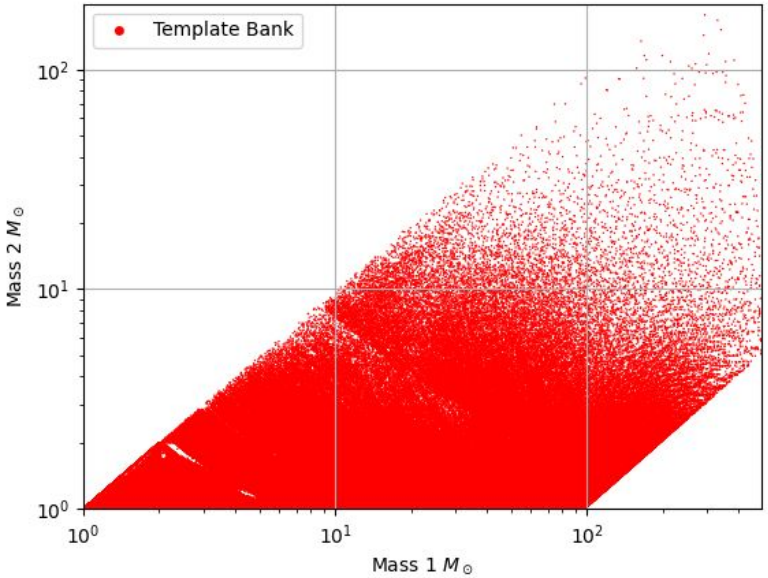
15 parameters

Degeneracy between various parameters

Family of Waveforms



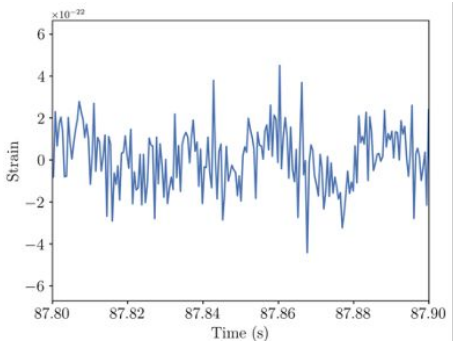
Template bank



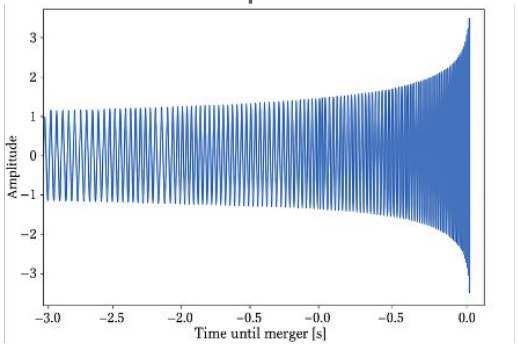
- 15 dimensional grid – computationally infeasible
- Various placement algorithms – stochastic / geometric / hybrid
- Neighbouring points must have a match greater than a threshold

Matched Filtering

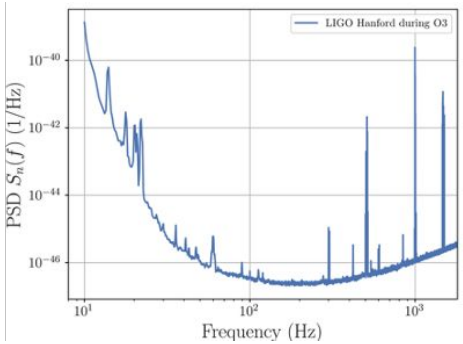
Data

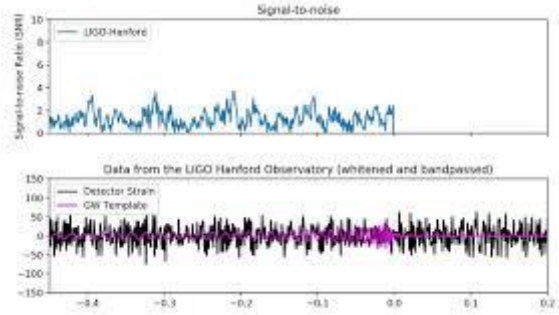


Template



Detector noise





$$(h|s) = 4\mathrm{Real}\int \frac{\tilde{h}^*(f)\tilde{s}(f)}{S_n(f)}df$$

Optimal statistic in **Gaussian** and **Stationary** noise

Hypothesis testing

Only noise

$$\mathcal{H}_0 : s(t) = n(t)$$

$$p(s|\mathcal{H}_0) \propto e^{-(s|s)/2}$$

Noise + signal

Best guess

$$\mathcal{H}_1 : s(t) = h(t) + n(t)$$

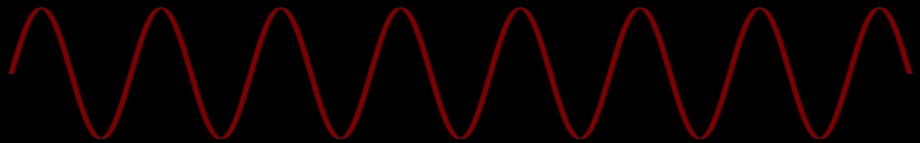
$$p(s|\mathcal{H}_1) \propto e^{-(s-h|s-h)/2}$$

Likelihood ratio

$$\Lambda(\mathcal{H}_1|s) = \frac{p(s|\mathcal{H}_1)}{p(s|\mathcal{H}_0)}$$

$$\log \Lambda(\mathcal{H}_1|s) = (s|h) - \frac{(h|h)}{2}$$

UNKNOWN PARAMETERS ?



Unknown amplitude

Observed signal

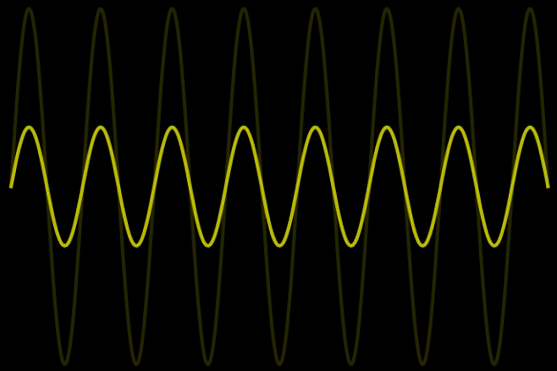
Reference signal (filter)

$$\tilde{h}(f;\mathcal{A}) = \mathcal{A} \times \tilde{g}(f)$$

$$\log \Lambda(\mathcal{H}_1|s) = (s|h) - \frac{(h|h)}{2}$$

$$= \mathcal{A}(s|g) - \mathcal{A}^2 \frac{(g|g)}{2}$$

$$\mathcal{A}_{\max} = \frac{(s|g)}{(g|g)}$$



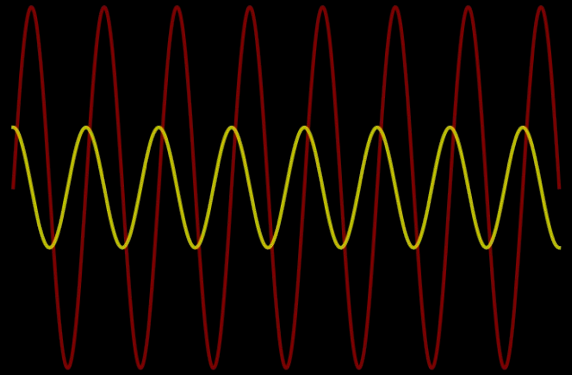
$$\log \Lambda = \frac{1}{2} \frac{(s|g)^2}{(g|g)} = \frac{\rho^2}{2}$$

Unknown phase

filters $(h|h) = (p|p) = (q|q)$

$$\tilde{h}(f;\theta) = \tilde{p}(f)\cos\theta + \tilde{q}(f)\sin\theta$$

$$\begin{aligned}\log\Lambda(\mathcal{H}_1|s) &= (s|h) - \frac{(h|h)}{2} \\ &= (s|p)\cos\theta + (s|q)\sin\theta - \frac{1}{2}(h|h) \\ &= x\cos\theta + y\sin\theta - \frac{1}{2}(h|h)\end{aligned}$$

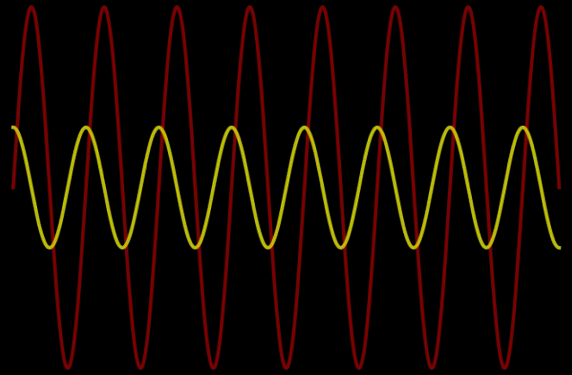


Unknown phase

filters $(h|h) = (p|p) = (q|q)$

$$\tilde{h}(f;\theta) = \tilde{p}(f)\cos\theta + \tilde{q}(f)\sin\theta$$

$$\begin{aligned}\log\Lambda(\mathcal{H}_1|s) &= (s|h) - \frac{(h|h)}{2} \\ &= (s|p)\cos\theta + (s|q)\sin\theta - \frac{1}{2}(h|h) \\ &= x\cos\theta + y\sin\theta - \frac{1}{2}(h|h)\end{aligned}$$



$z^2 = x^2 + y^2 \qquad \phi = \tan^{-1}(y/x)$

$$= z\cos(\phi - \theta) - \frac{1}{2}(h|h)$$

Maximizes when

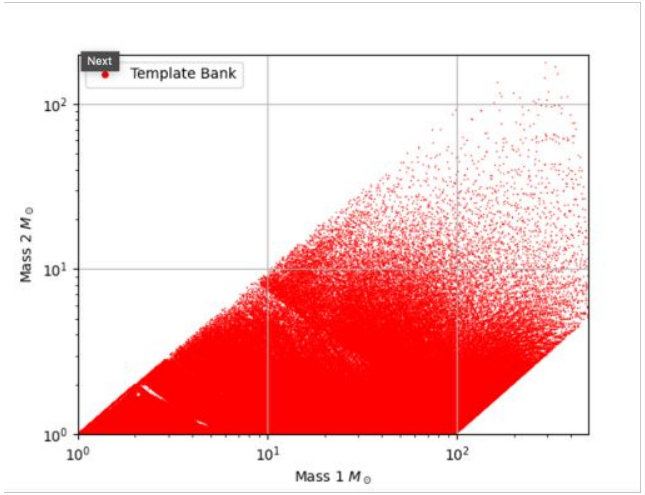
$$z^2 = (s|p)^2 + (s|q)^2$$

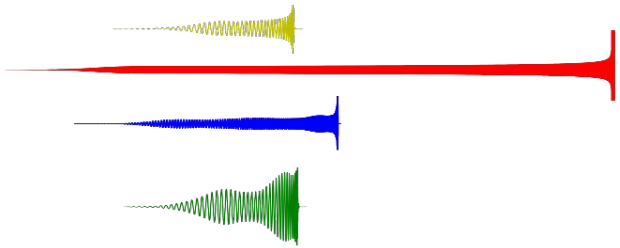
Search assumptions

- No precession
- No orbital eccentricity
- GW emission via dominant mode (l,m) = (2,2)

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

Maximising SNR over unknown parameters



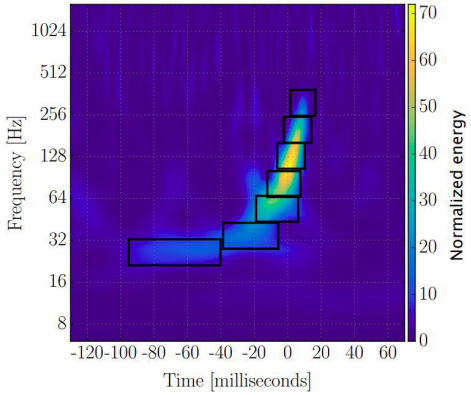


Masses and spins
(z-component)
4 dimensional

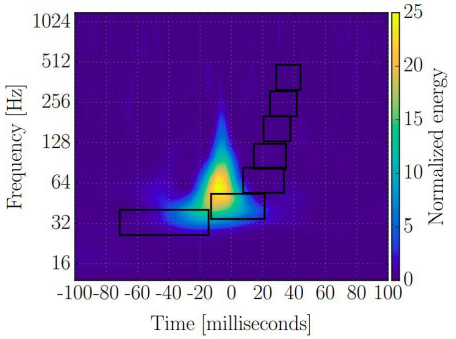
All the rest parameters

$$\rho = \sqrt{(\hat{h}_0|s)^2 + (\hat{h}_{\pi/2}|s)^2}$$

How do we handle non-stationarity ?



$\chi_r^2 \sim 1$



$\chi_r^2 > 1$

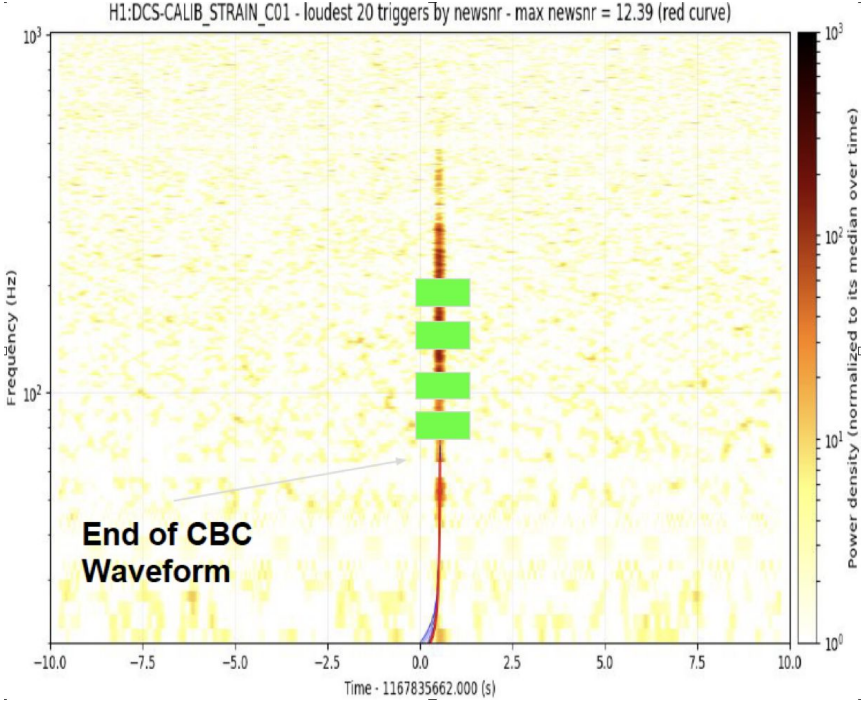
$$\chi_r^2 = \frac{p}{2p-2}\sum_{i=1}^{p}(\text{expected} - \text{observed})^2$$

$$\hat{\rho} = \frac{\rho}{[(1+(\chi_r^2)^3)/2]^{1/6}}$$

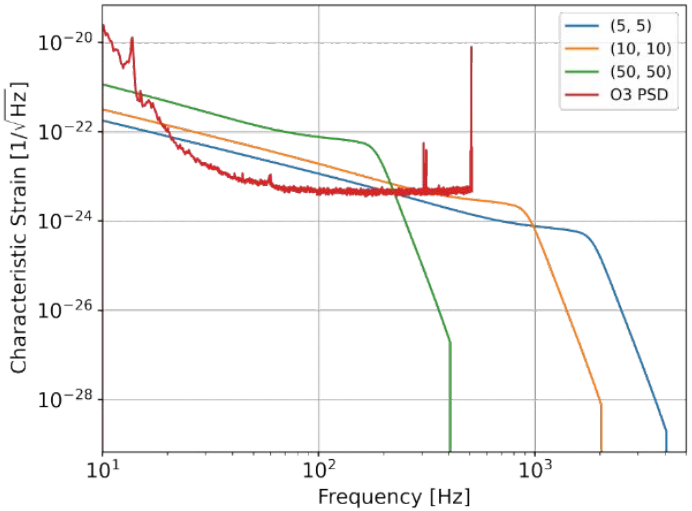
newSNR

Courtesy: Laura Nutall

Another chi-squared test



Courtesy: Ian Harry



We expect less power at high frequencies

Signal consistency checks

Coincidence

- Look for triggers with the same template in different detectors
- Require timing difference between detectors to be less than light travel time
- Rank coincident triggers based on log-likelihood of being a real signal

$$R = \log(r_S(\vec{k})) - \log(r_N(\vec{k}))$$

- Ranking statistic requires modeling trigger rates due to signal and noise

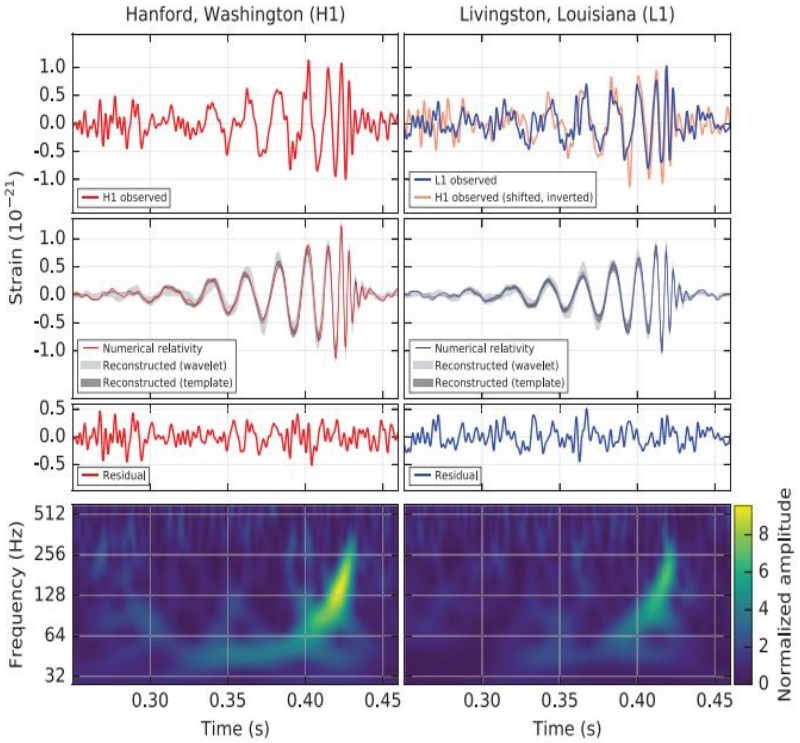


Image taken from B. P. Abbott et al 2016

The Signal Model

The overall rate of signals is constant.

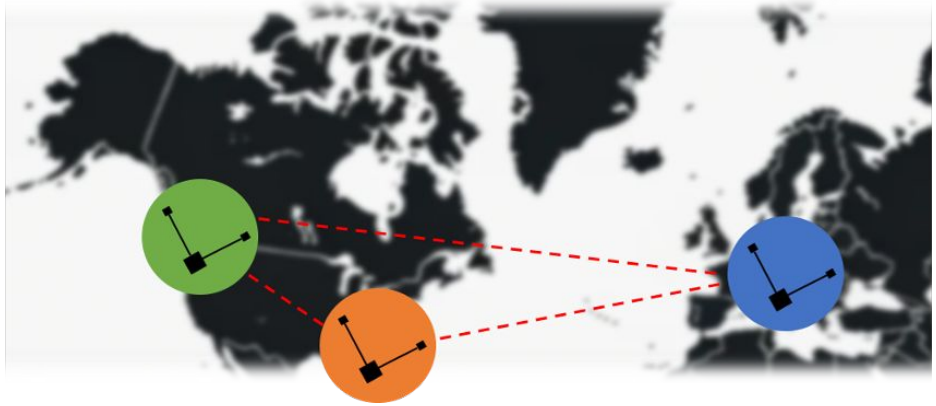
- Normalized density of signals over parameter space
- Stored as a KDE

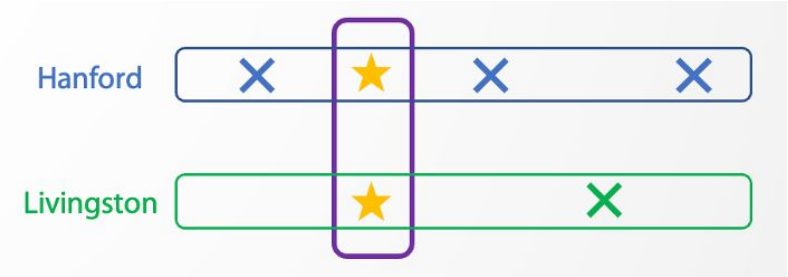
$$r_S = \mu_S \, \sigma^3(\vec{\theta}) \, d_S(\vec{\theta}) \, p_S(\vec{\Omega})$$

- Tracks variations in sensitive volume
- Normalized for a reference PSD

- Probability of phase, time and amplitude differences for signals
- Stored as a histogram

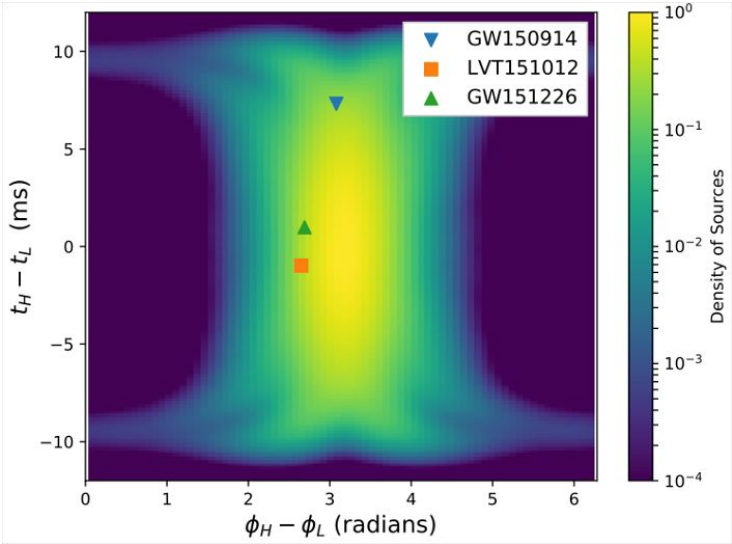
Consistency between multiple detectors





15 ms time window

Expectations from “real” signal triggers



DOI *10.3847/1538-4357/aa8f50*

The Coincident Noise Model

- Normalized density of ***templates*** over parameter space
- Stored as a KDE

Product is taken over detectors

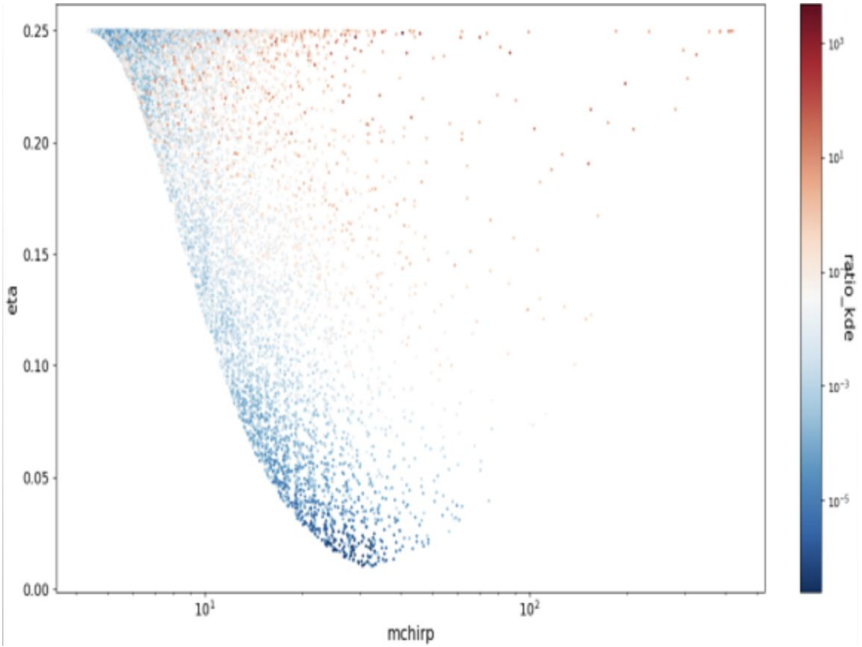
$$r_N = d_T(\vec{\theta})\, A_{\{d\}} \prod_d r_d(\rho_d, \vec{\theta}, t)$$

The timing window for coincident triggers in a set of detectors {d}

- Rate density of triggers as a function of SNR
- Allowed to vary over parameter space and over time
- Needs more modeling

KDE-based ranking

- Triggers from every template is **not** equally likely
- Account for signal / templates distribution accurately
  - Higher density of templates at low mass
  - Higher rate of BBH signals than BNS / NSBH



Courtesy: Praveen Kumar, Tom Dent

The Single Detector Noise Model

We model the rate density of triggers above a threshold SNR ρ_{th} as a decaying exponential with a time dependent correction factor

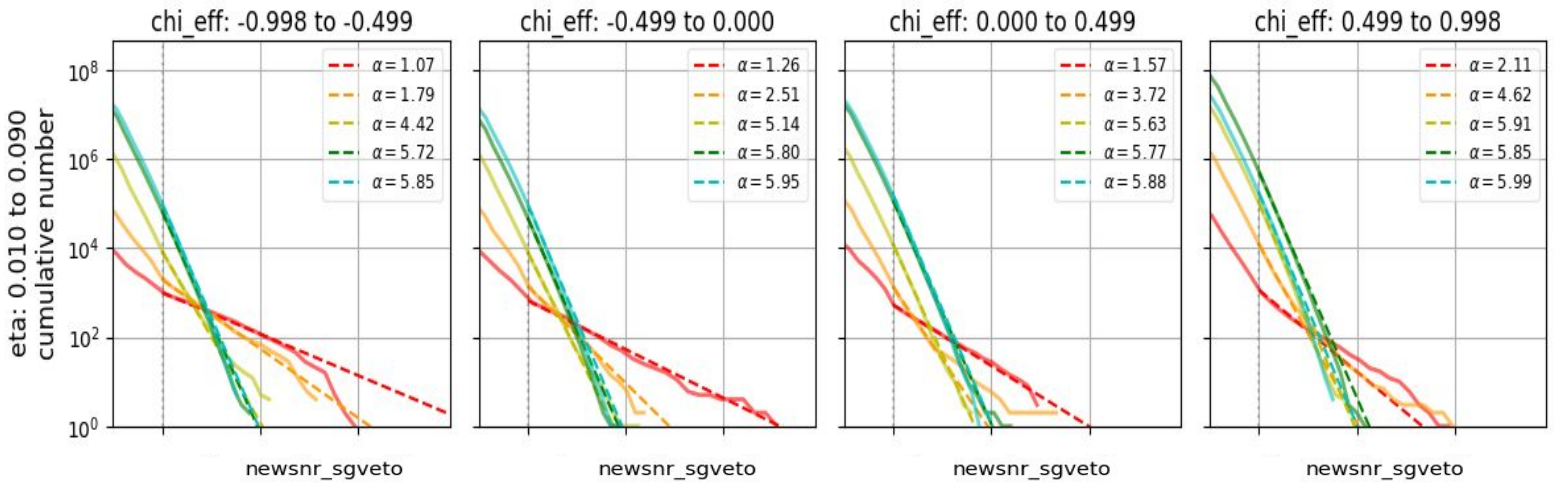
- Time dependent correction factor based on data quality information
- Needs even more modeling...

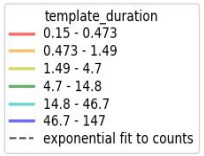
$$r_d = \mu(\vec{\theta})\,\delta(\vec{\theta}, t)\,\alpha(\vec{\theta})\exp\left(\alpha(\vec{\theta})\,(\rho - \rho_{th})\right)$$

The overall rate of triggers in template $\vec{\theta}$ with $\rho > \rho_{th}$

Exponential fit coefficient is fit for each template, then smoothed over nearby templates

Single detector ranking





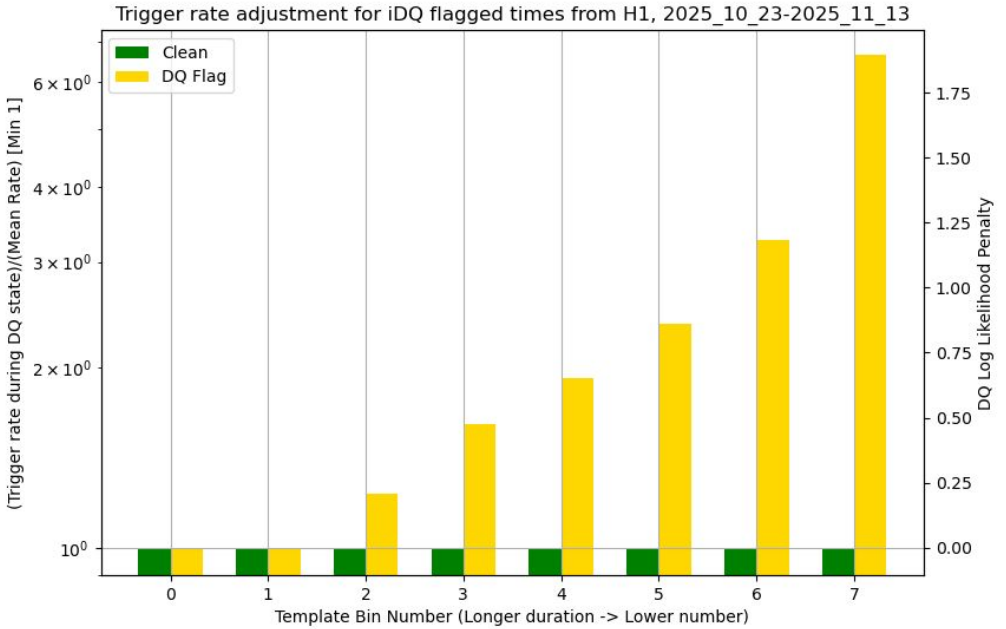
SNR ✚ Chi-square ✚ Sine-gaussian chi-square

Modeling Time-dependence with Data Quality Flags

- Use data quality flags to divide observing time into flagged and unflagged segments
- Divide templates into bins based on duration
- For each template bin, calculate correction factors by comparing trigger rates in flagged times to the mean trigger rate

$$\delta_{flag} = \frac{N_{flag}/T_{flag}}{N_{total}/T_{total}}$$

- Also calculate correction factor for unflagged times in equivalent way
- Correction factors are restricted to be >= 1 to avoid artificial upranking



Single detector ranking

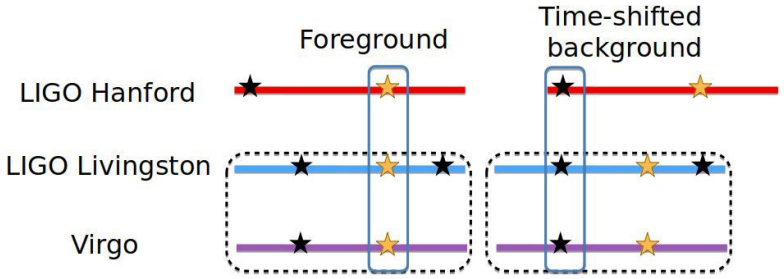
- We also look for signals in a single detector
- Ranking statistic for single-detector candidates is formed by dropping all the coincidence dependent terms from the coincident ranking stat
- Generally less sensitive than the coincident search due to higher noise rates

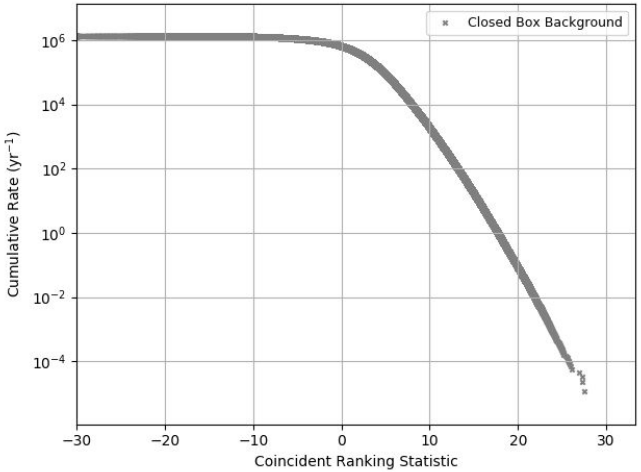
$$r_S = \mu_S \, \sigma^3(\vec{\theta}) \, d_S(\vec{\theta})$$

$$r_N = d_T(\vec{\theta}) \, r_d(\rho, \vec{\theta}, t)$$

False Alarm Rates

- False alarm rate measures how often noise triggers are ranked higher than a candidate event
- For coincident triggers, FARs are calculated using time slides
- For single detector triggers, use an exponential fit of the ranking statistic
- Hierarchically remove highest ranked triggers from background to avoid bias from real signals

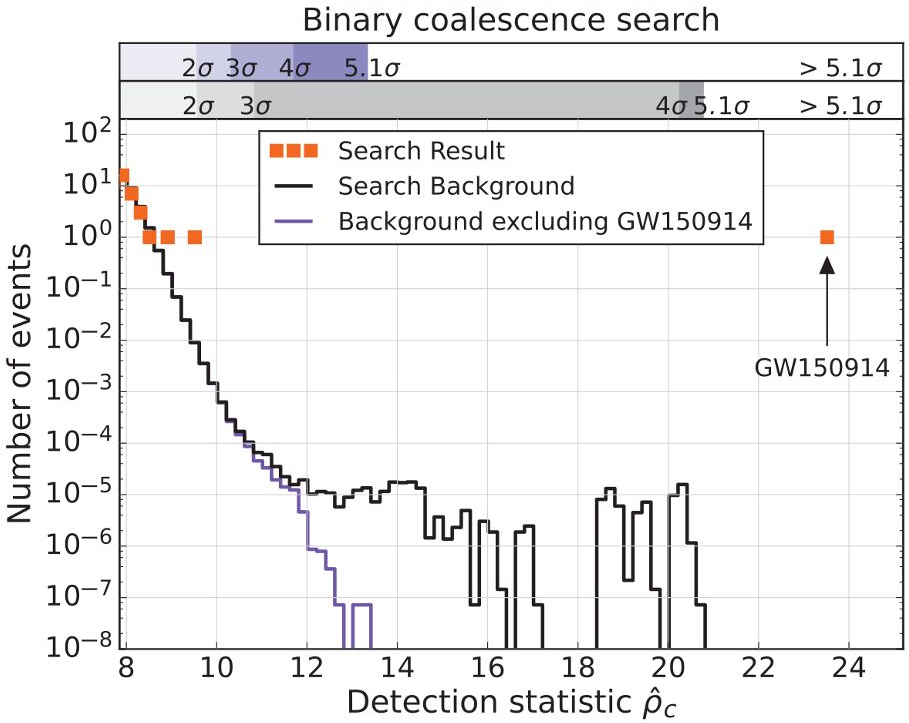




Top image from Gareth Cabourn-Davies et al. 2020

Is it a detection ?

False alarm rate



FAR ~ 1 in Million years

PhysRevLett.116.061102

PyCBC in Action: GW200129

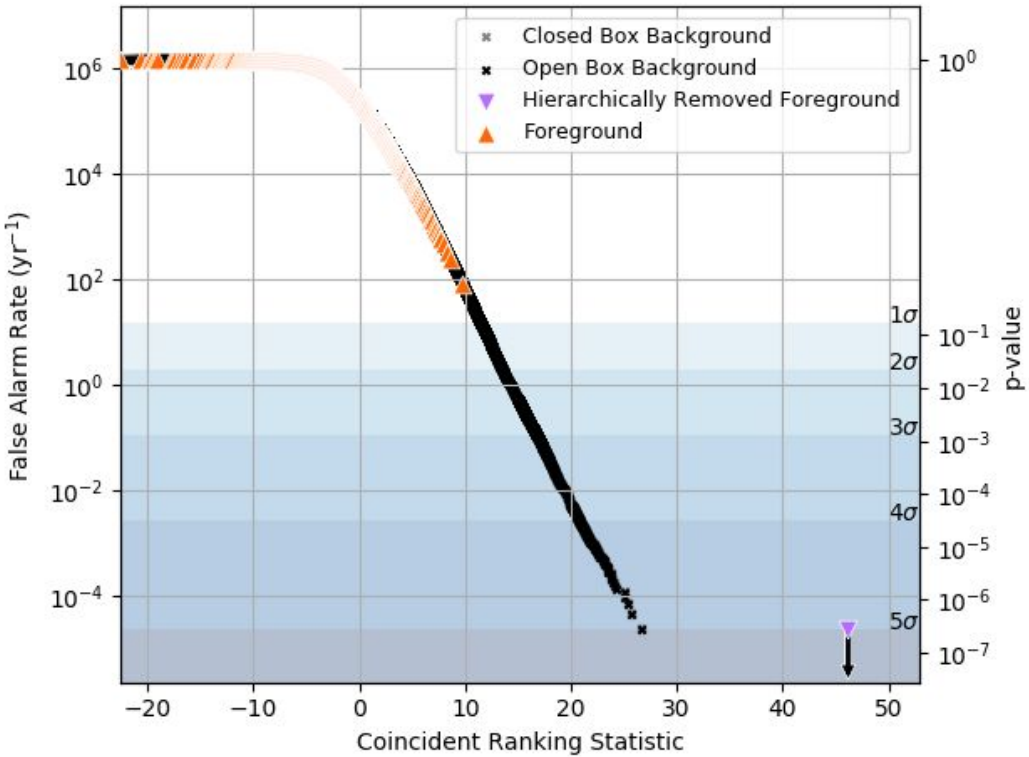
- GW found by search with ranking statistic 46.10
- False alarm rate 1 per 42606.90 years
- CBC with component masses

$$M_1 = 34.5^{+9.9}_{-3.2} M_\odot$$

$$M_2 = 28.9^{+3.4}_{-9.3} M_\odot$$

- Luminosity distance

$$D_L = 0.90^{+0.29}_{-0.38} \mathrm{Gpc}$$



PyCBC - Lecture 2

Rahul Dhurkunde, Max Trevor

Deep Offline Search

Template bank

Parameters	Values
spin ranges	$\chi_{\rm BH} \leq \pm 0.997, \chi_{\rm NS} \leq \pm 0.05$
NSBH boundary mass	$m_{\rm NSBH} = 2M_\odot$
Total mass	$2M_\odot \leq M \leq 500\mathrm{M}_\odot$
Mass ratio	$1 \leq m_1/m_2 \leq 97.989$
Lower frequency	$f_{\rm low} = 15\mathrm{Hz}$ if $M > 100\mathrm{M}_\odot$ else $f_{\rm opt}$
Minimum template duration	70 ms
Optimize $f_{\rm low}$	0.995
Dynamic minimal match	$\mathcal{MM} = 0.98$ if $M_c > 7.5$ else $\mathcal{MM} = 0.965$
Waveform model	`SEOBNRv5_ROM`

~ 700, 000 templates

- Configuration
  - Waveform : `SEOBNRv4_ROM`
  - Full ranking statistics : `dq_phasetd_exp_fit_fgbg_kde`
  - Single detector candidates are also identified

Where do we perform offline searches ?

- Computing sites
  - Open Science Grid
  - High throughput: 2GBs memory, 6-7 hours running up to **10 Million** jobs.
  - Post processing is done locally on CIT.

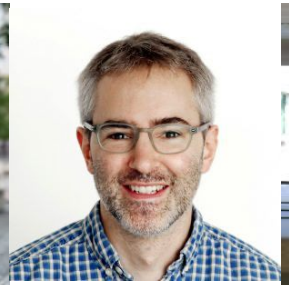
How do we package offline searches ?

- Scheduling
  - Scheduled via **HTCondor**.
  - Each job will have its own *submit*, *error* and *out* file.
  - **Pegasus** for packaging the workflow.

How do we pass inputs to offline searches ?

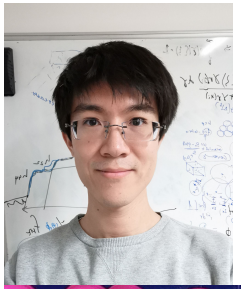
- Data storage
  - GW data is fetched stored on cluster or is pulled from CVMFS / OSDF.
  - Use singularity images for our environment.
  - Fast read access on OSG sites via OSDF.

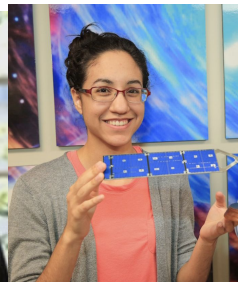
PyCBC team within LIGO





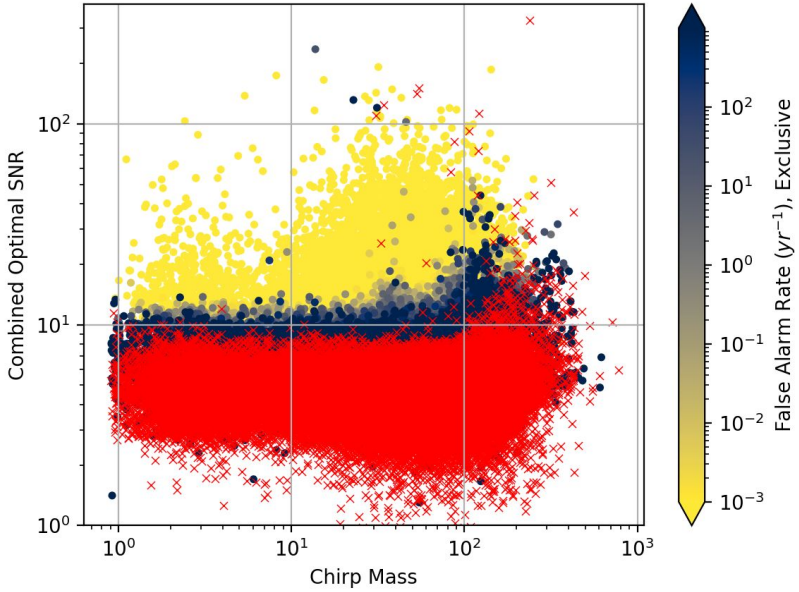




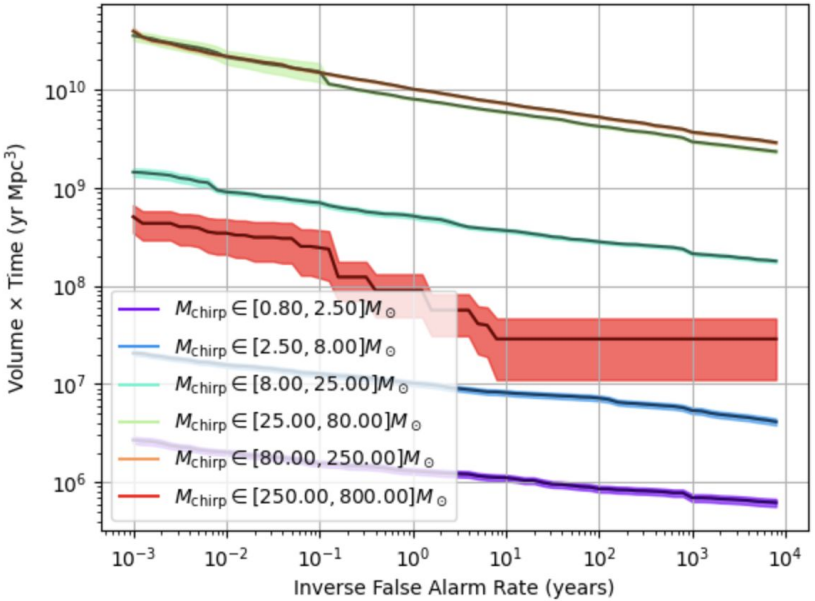

….. And at least 20 more active members

Search sensitivity



- Use astrophysically distributed population of simulated injections.
- Fraction of recovered signals $\propto$ sensitive <*VT*>

GWTC 4.0 using PyCBC



> Not all of the 129 candidates with $p_{\mathrm{astro}} \geq 0.5$ were observed with $p_{\mathrm{astro}} \geq 0.5$ by all pipelines. Of these candidates, 54 were found by cWB-BBH, 89 were found by GstLAL, 76 were found by MBTA, and 103 were found by PyCBC. Only 39 of these candidates were found by all

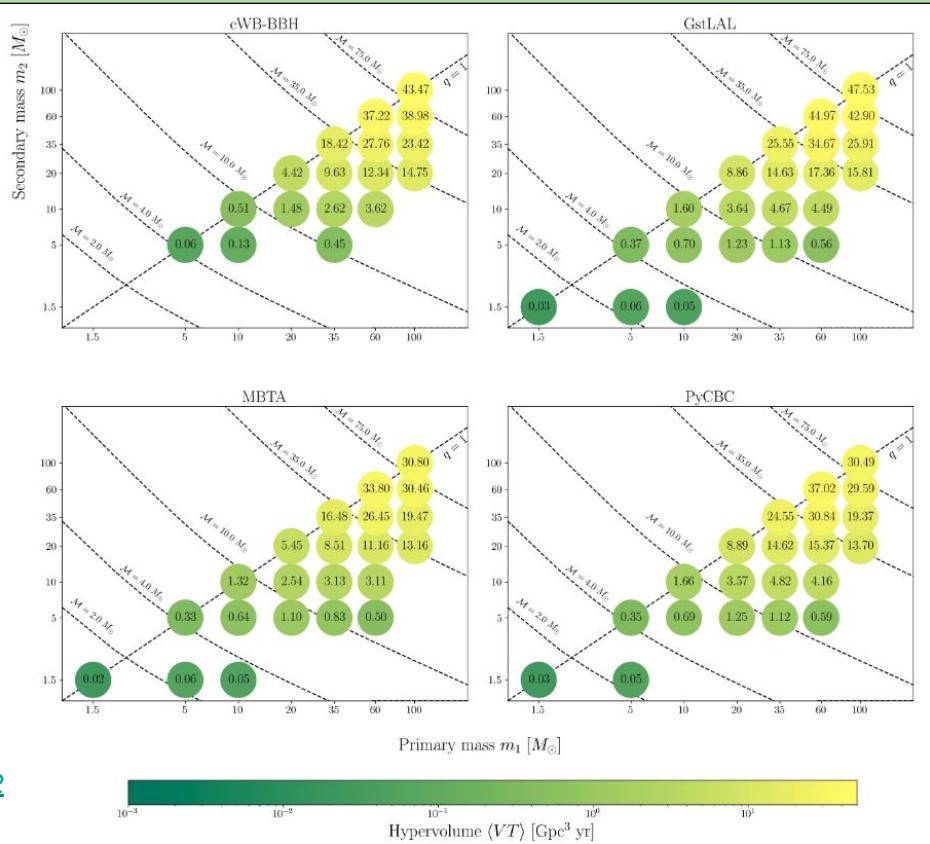
GWTC 4.0 Methods paper

Notice the drop in sensitivity for high mass systems

Comparison with other pipelines

PyCBC has **good** sensitivity at ~ 30 – 30 Msun. But **not** for IMBH systems.

Searches are complementary.



GWTC 4.0 results paper: arXiv:2508.18082

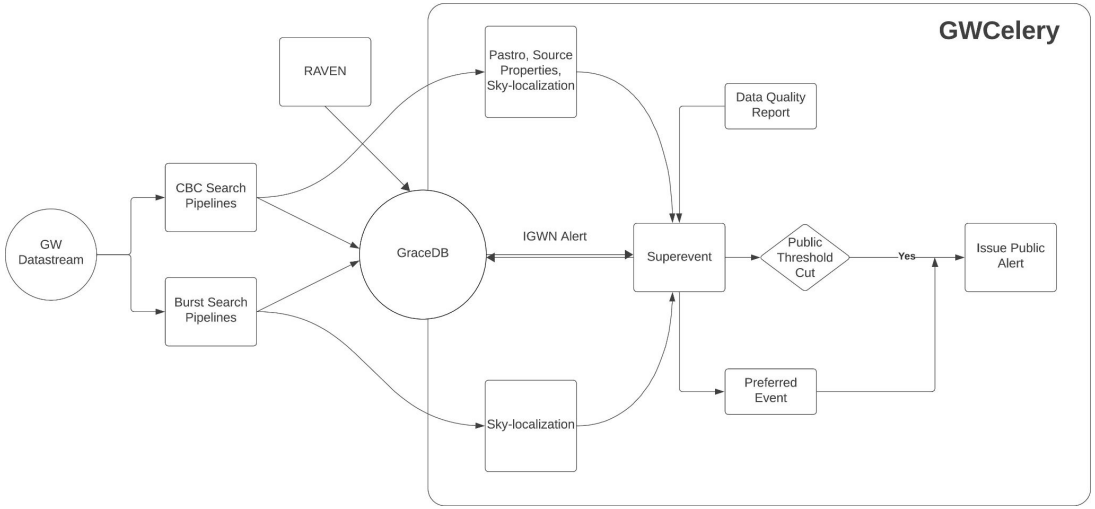
PyCBC Live

PyCBC Live

- PyCBC Live is the low-latency version of PyCBC
- Doesn’t use HTCondor, instead uses worker processes that run continuously and communicate through MPI
    - Downside of MPI is that problems with one worker process can kill the whole search
- In O4, used 150 worker nodes, each filtering O(1000) templates
- Read data in 8 second chunks, filter and produce results before the next 8 seconds of data becomes available
- End to end latency (from data taking at IFO to event upload) is ~15 sec

Low-latency operations

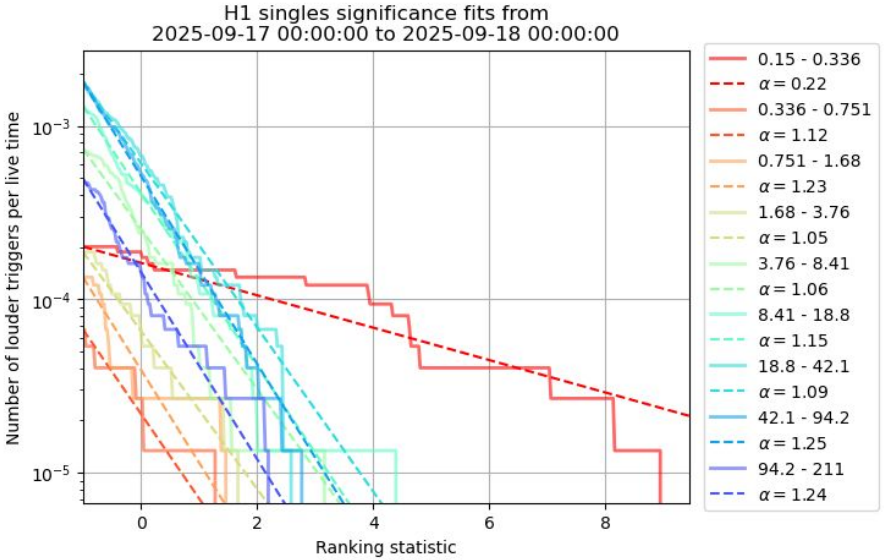
- PyCBC Live is one just one piece of the larger low-latency alert pipeline
- Other pieces include:
  - Data calibration and distribution
  - iDQ for automated glitch prediction
  - Other search pipelines (CWB, GstLAL, MBTA)
  - Raven for reporting coincident events from neutrinos or electromagnetic astronomy
  - Data quality reports for vetting event candidates
  - Parameter estimation to refine estimates of source properties
  - GraceDB is a web server for gathering event information in one place
  - GWCelery for event annotation and alert distribution



- It takes all of these pieces (and more) working together to detect gravitational waves and alert the multimessenger astronomy community

Ranking triggers online

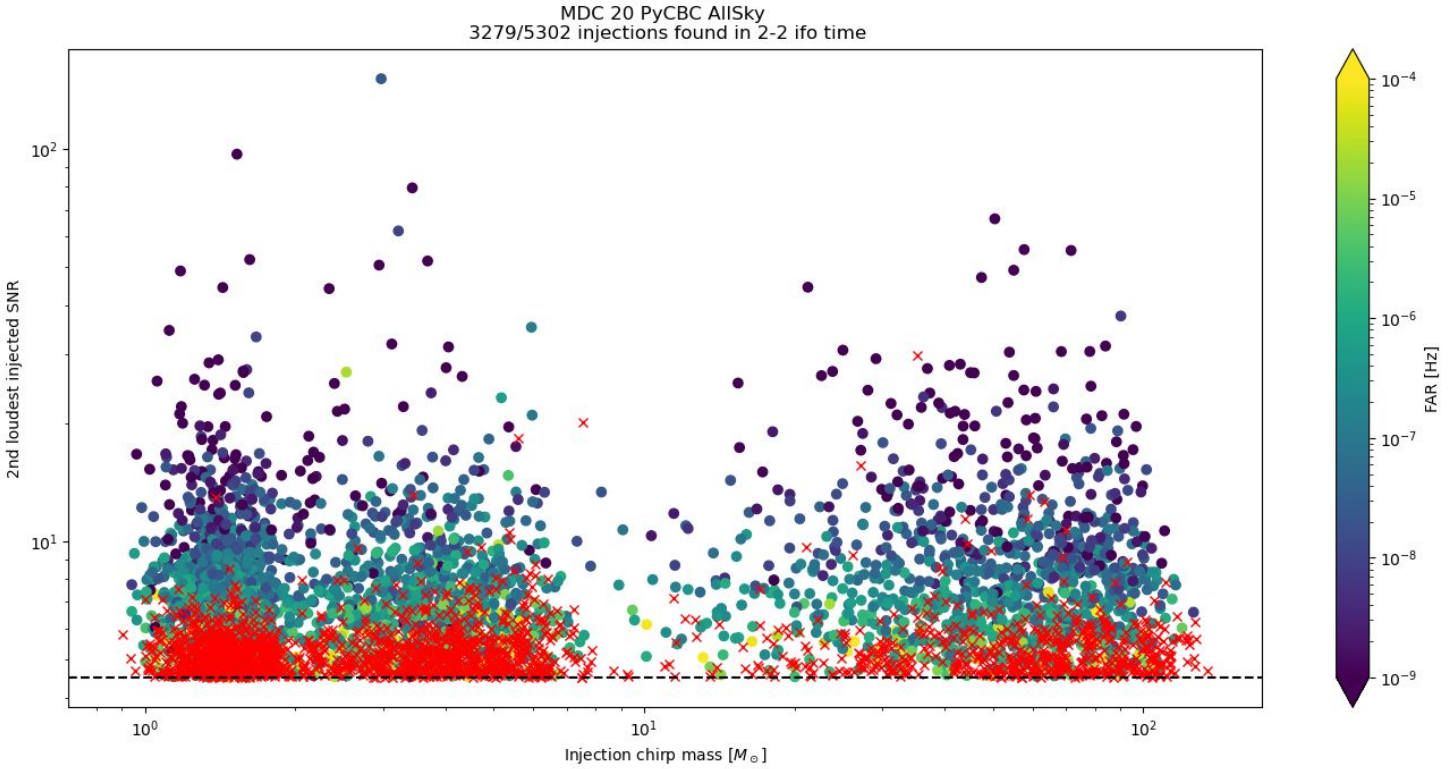
- Compared to offline search, PyCBC Live needs to use much less data at any given time to keep computation lightweight
  - Only store ~5 hours of h(t) from each detector, limiting FAR to 1/100 years
- At the end of each day, perform background model fits to that day's data
- 20 days worth of daily fits are combined to construct background used by the search



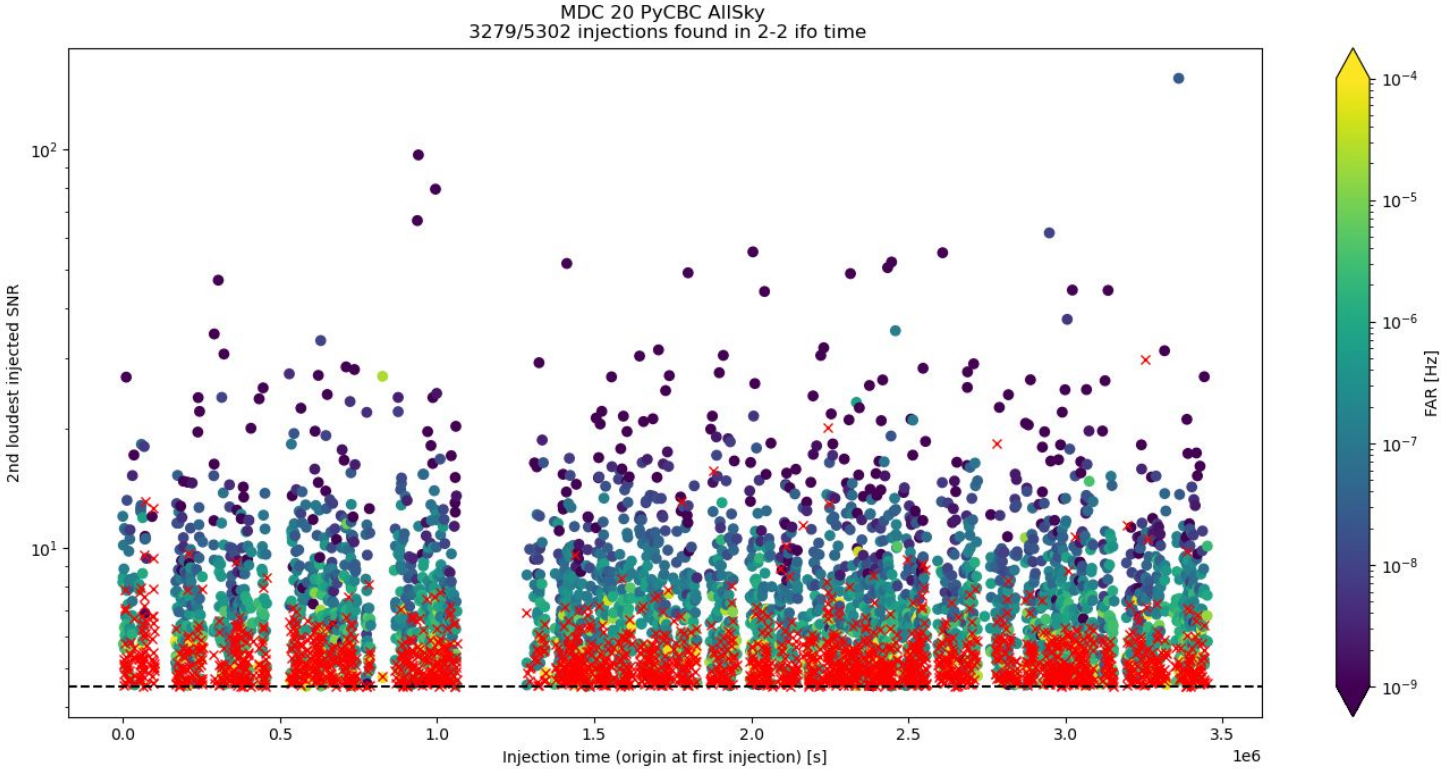
Testing PyCBC Live

- We regularly test the performance of PyCBC Live using mock data challenges, including the Low-Latency Production Integration Challenge (LLPIC)
- A mock data challenge replays weeks of real interferometer data, but with thousands of simulated signals injected into the data
- Allows for detailed analysis of search sensitivity to different types of events
- Compare performance across multiple replays to evaluate effect of changes to the search
- Current technical issue: The high rate of signals pollutes the background model, increasing reported FARs

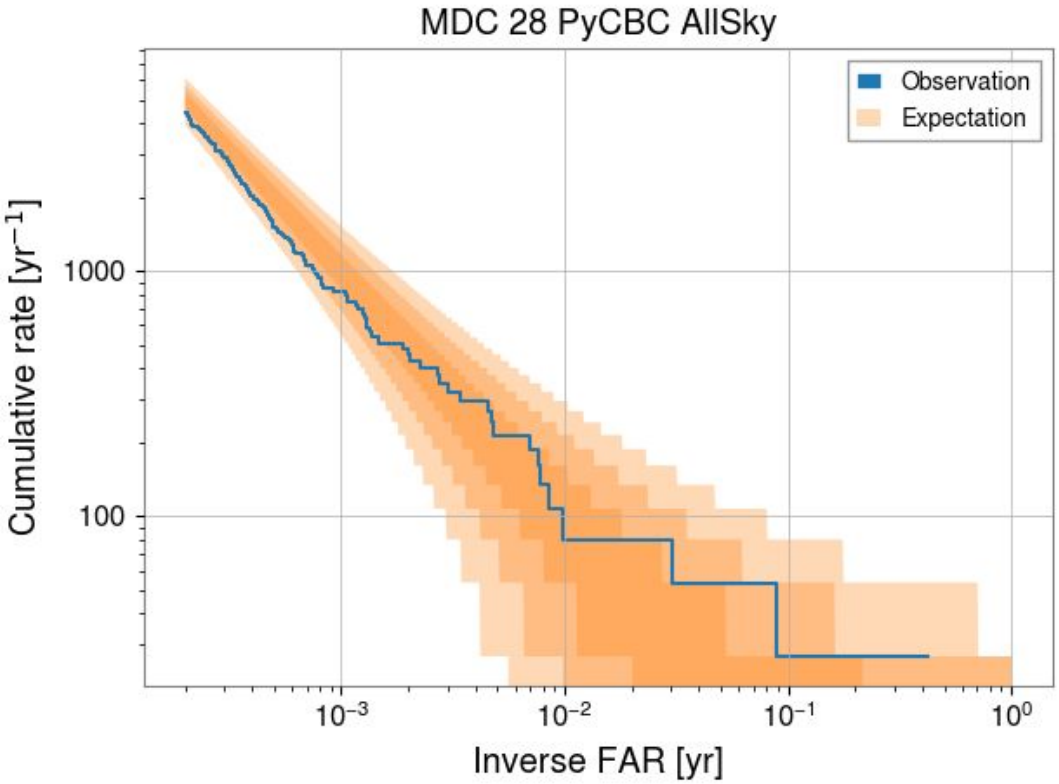
MDC Results


MDC 20 PyCBC AllSky
3279/5302 injections found in 2-2 ifo time
2nd loudest injected SNR
Injection chirp mass [$M_\odot$]
FAR [Hz]
10^2
10^1
10^0
10^1
10^2
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}

MDC Results


MDC 20 PyCBC AllSky
3279/5302 injections found in 2-2 ifo time
2nd loudest injected SNR
10^2
10^1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Injection time (origin at first injection) [s]
1e6
FAR [Hz]
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}

MDC Results


MDC 28 PyCBC AllSky
Observation
Expectation
Cumulative rate [yr^{-1}]
1000
100
10^{-3}
10^{-2}
10^{-1}
10^{0}
Inverse FAR [yr]

Bonus Content: Detector Characterization

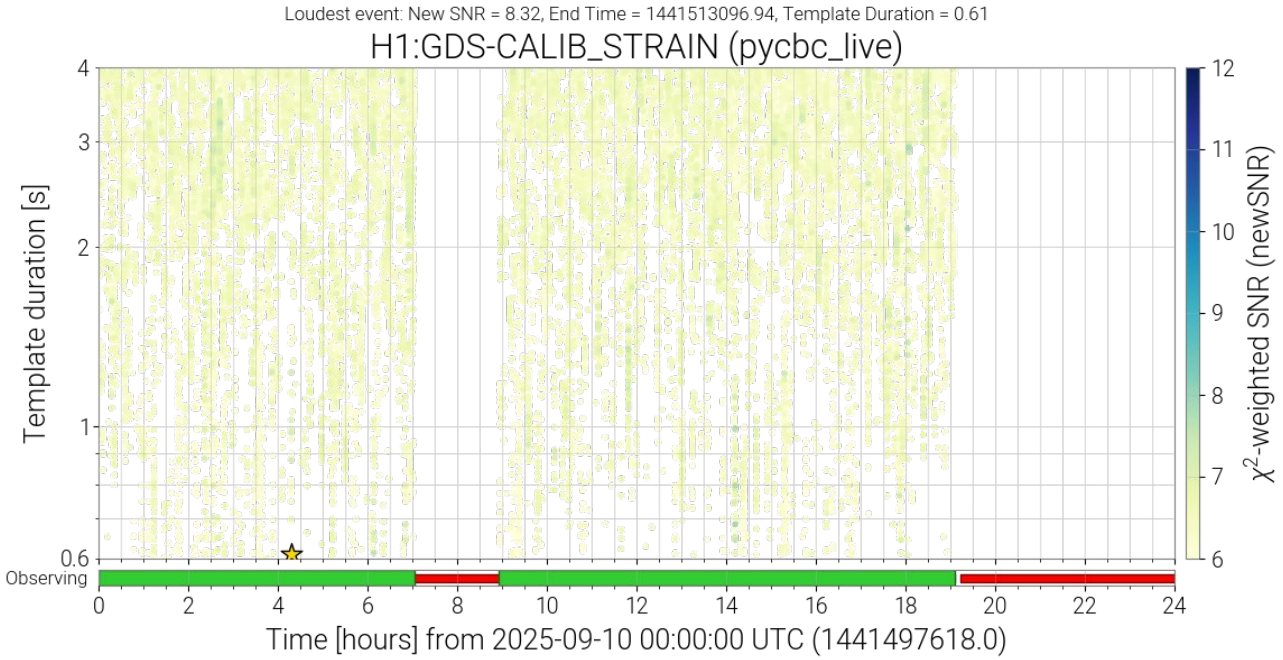
- Detector characterization seeks to understand and remove sources of excess noise in the detector
- Search pipelines assume that data is *stationary* and *gaussian*
  - Violating either assumption can lead to false positives
- For a search pipeline, it is useful to know what times may have nonstationary or nongaussian data
  - In PyCBC, we use the time-dependent portion of the noise model to account for differences in trigger properties during flagged times
- DetChar maintains a variety of tools to monitor, predict, and model data quality problems
- LIGO summary pages allow monitoring of data quality in real time:
  - Available at summary.ligo.org

Detector Characterization cont.

From the summary page monitor of PyCBC Live triggers, here is a good day of data at LIGO Hanford:

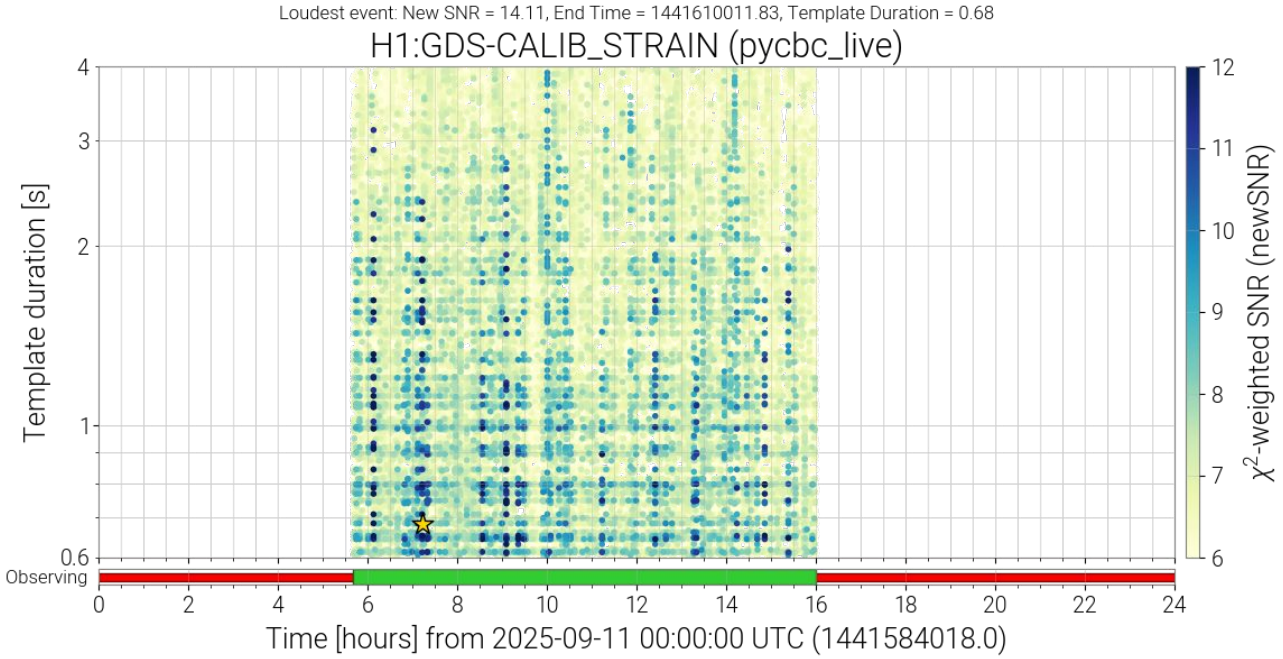
Colored dots indicate SNR of triggers at a particular time, with a particular template duration

Green and red bars at the bottom indicate when the detector was observing



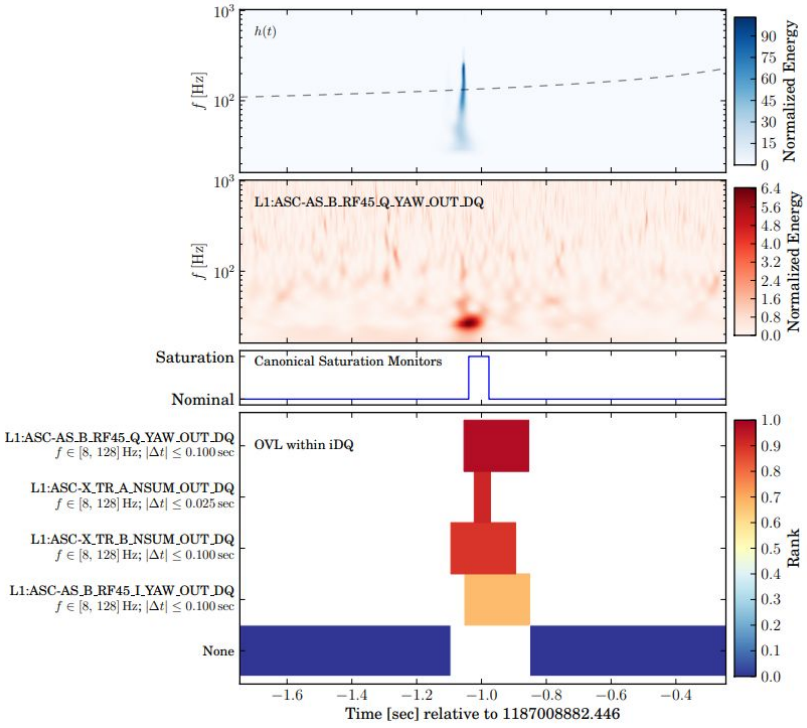
Detector Characterization cont.

Here is a really bad day of data:



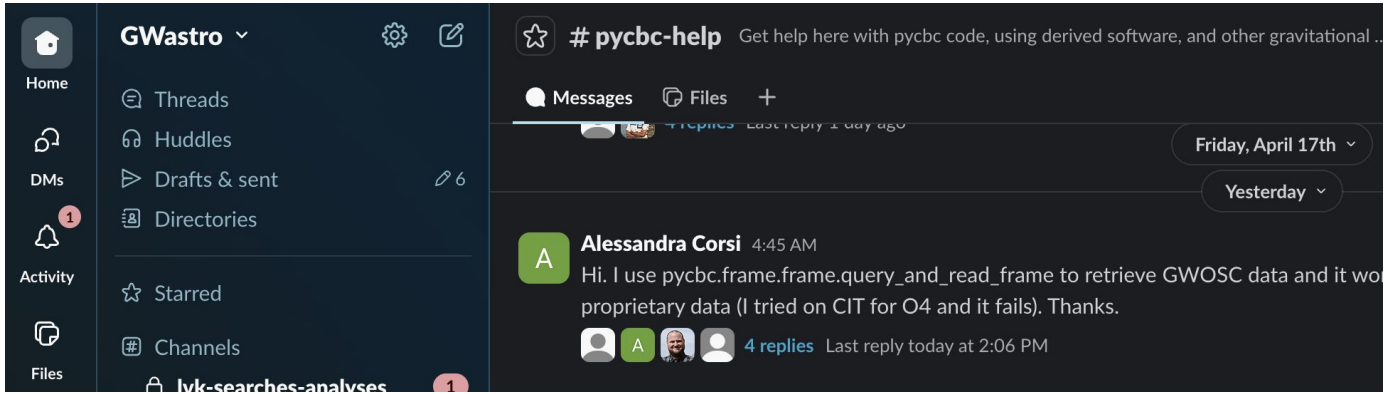
Bonus Pipeline: iDQ

- iDQ produces automated measures of data quality
- Relies on a trigger generator (Omicron or SNAX) to find times of excess power in auxiliary channels and in h(t)
- iDQ correlates triggers in auxiliary channels with glitches in h(t)
- Produces a timeseries indicating the probability of a glitch at each sample in time
- For PyCBC, we produce data quality flags by thresholding on glitch probability
- iDQ is in dire need of more personpower to support development and maintenance
    - To get involved, contact Zach Yarbrough



Areas of development

- Missing physics
  - Precessing binaries
  - Eccentric binaries
  - Hyperbolic orbits
- Better glitch rejection
  - Automate production of data quality flags
  - Improve autogating
  - Don’t assume that slope of power law fit stays the same during flagged times
- Faster and more robust searches
  - Improve low-latency background fitting
  - Remove confident detections from background estimation
  - Explore use of GPUs in production


GWastro
Home
DMs
Activity
Files
Threads
Huddles
Drafts & sent
6
Directories
Starred
Channels
pycbc-help
Get help here with pycbc code, using derived software, and other gravitational ..
Messages
Files
Friday, April 17th
Yesterday
A
Alessandra Corsi
4:45 AM
Hi. I use pycbc.frame.frame.query_and_read_frame to retrieve GWOSC data and it wo
proprietary data (I tried on CIT for O4 and it fails). Thanks.
4 replies
Last reply today at 2:06 PM