



Searching for Gravitational Waves from the Coalescence of High Mass Black Hole Binaries

LIGO SURF 2016 Final Presentation
August 18, 2016

Ka-Lok (Rico), Lo

Department of Physics, The Chinese University of Hong Kong
Mentors: Surabhi Sachdev, Kent Blackburn, and Alan Weinstein

LIGO Laboratory, California Institute of Technology

Overview

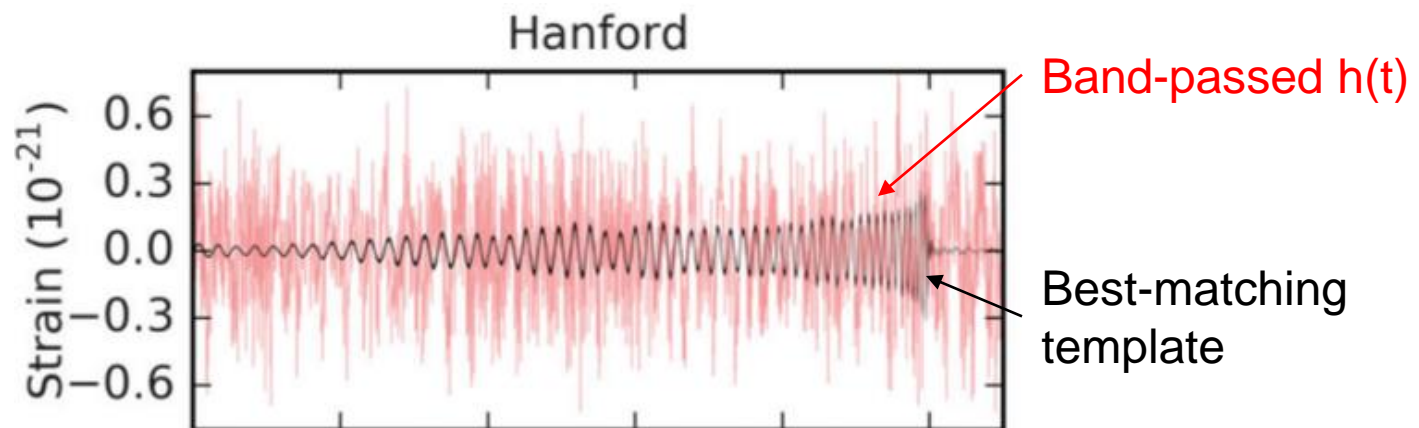
- Introduction
- Tuning Search Parameter: f_{\min}
- $h(t)$ Gating
- Other Projects
- Acknowledgement
- Reference
- Q&A

Introduction

1. Background
 - Searching for Gravitational Waves
 - Matched Filtering
 - gstlal: the search pipeline
2. Objectives of the research project

Searching for Gravitational Waves

- Amplitude of GW is small (10^{-21} to 10^{-28})
- Even worse, the detector is noisy
- Impossible to detect one by naked eye
- Need a more reliable and systematic way to search for GWs (proper data analysis)



Matched Filtering

- A powerful technique to search for a signal from a noisy data if **we know what we are looking for**
- Procedures:
 1. Figure out the expected waveforms (a lot of physics!)
 2. Generate many of those expected waveforms (*templates*), forming a *template bank*
 3. Slide a template across the signal and compare the 'similarity'
 4. Find peak of the matched filter responses
 5. Repeat Step 3 and 4 for all the templates in the bank
 6. Find coincident triggers (found in both H1 and L1)
- Template bank defines the parameter space (*masses and spins*) covered by a search

Matched Filtering: Definition of SNR

- Q: How can one quantify the similarity between the template and signal?
- A: Compute the **Signal-to-Noise Ratio (SNR)**
- Suppose we are filtering the data $d(t)$ with some normalized templates $\{h_i^c(t)\}$

$$z_i(t) = x_i(t) + iy_i(t)$$

Matched filtering response to $h_+(t)$

$$= 4 \int_0^\infty df \frac{\tilde{h}_i^{c*}(f) \tilde{d}(f)}{S_n(f)} e^{2\pi ift}$$

Matched filtering response to $h_x(t)$

$$SNR = \rho = |z|$$

Power Spectral Density of Noise

gstlal: the search pipeline

- Low Latency (on-line) search pipeline for compact binary coalescences (CBC)
- EM Follow-up made possible (e.g. X ray, Visible light, Radio wave emitted from BNS merger)
- Average latency \sim 1 minute (median latency)
- Can be operated in either on-line or off-line mode
- Based on matched filtering and detector network coincidence

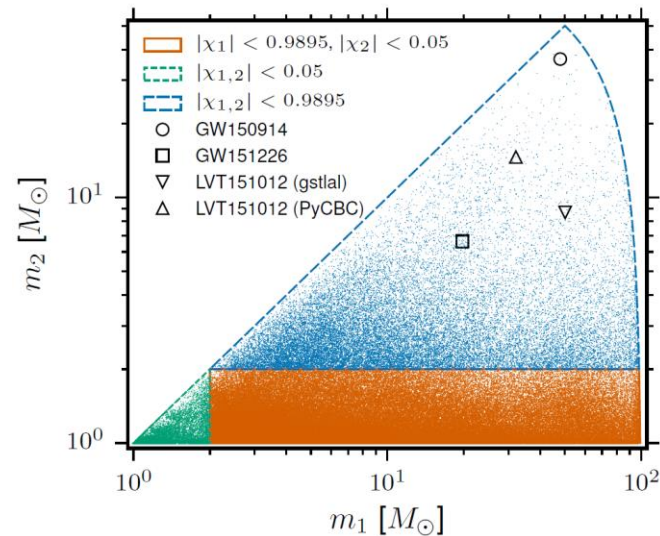
Parameter Space Covered by gstlal (uberbank)

- Currently

$$m_1, m_2 \geq 1M_{\odot}$$

$$2M_{\odot} \leq m_1 + m_2 \leq 100M_{\odot}$$

- Assumed component spins aligned with orbital angular momentum (no precession)



Objectives of this project

- In short: **optimizing the search**
- More specific:
 - Expanding the parameter space to cover Intermediate Mass Black Hole (IMBH) region

$$m_1, m_2 \geq 1M_{\odot}$$

$$2M_{\odot} \leq m_1 + m_2 \leq 600M_{\odot}$$

- But there are a lot more to consider

What can possibly go wrong?

- For instance:
 1. Change in sensitivity for different astrophysical sources
 2. Change in performance and extra computational costs (important for **real-time** GW detections)
 3. Ensuring high mass templates are sensible
 4. Compatibility of existing codes
- We want (or try) to answer them!

Tuning Search Parameter:

$$f_{\min}$$

1. Motivation: Why do we need to tune f_{\min} ?
2. Results
 - 20 Hz vs. 30 Hz
 - 15 Hz vs. 30 Hz
 - 15 Hz vs. 20 Hz

Observing GW Signals from IMBH

- Higher the mass of the binary system, lower the ring-down frequency

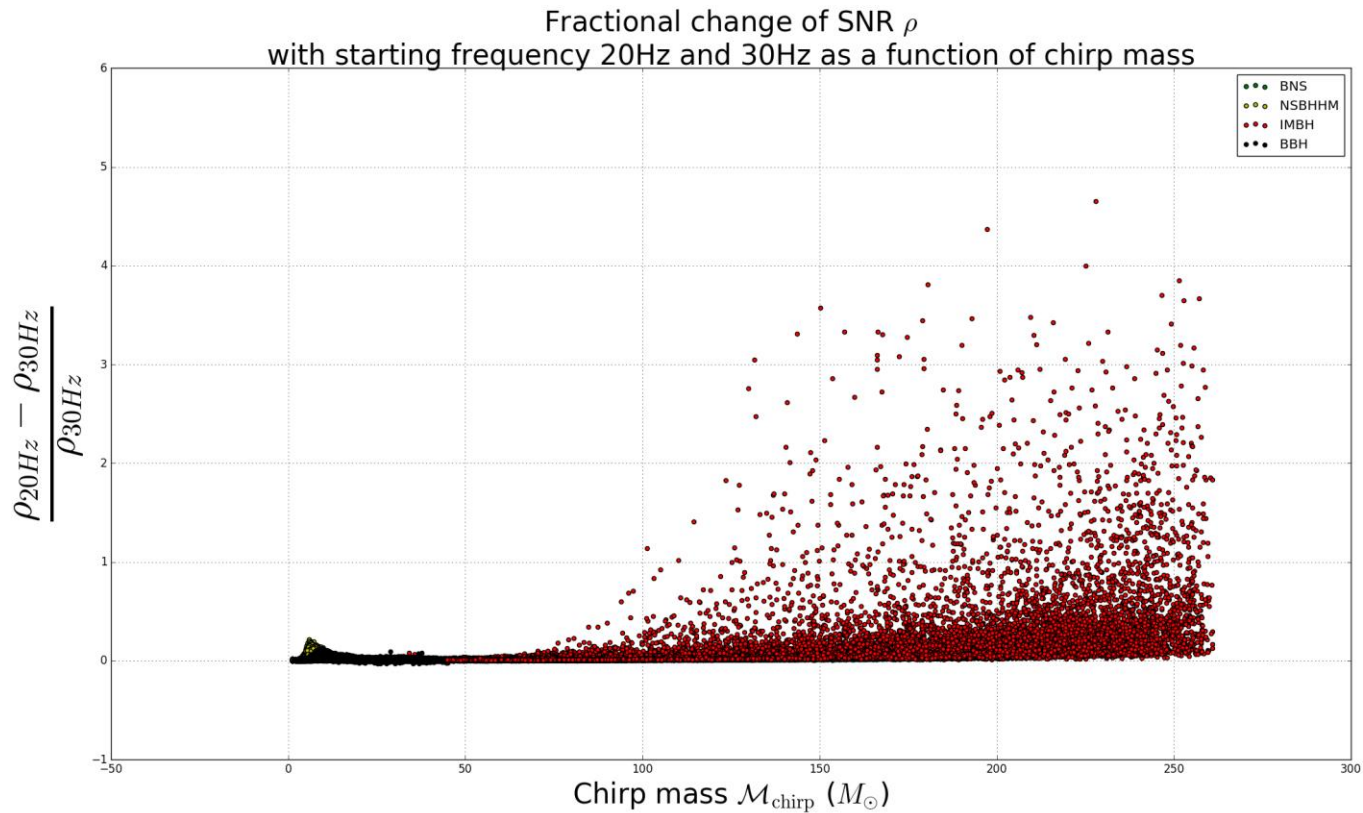
$$f_{ringdown} \approx \left[1.5251 - 1.1568(1-j)^{0.1292} \right] \frac{c^3}{2\pi GM}$$

- M is the total mass and j is the reduced Kerr spin parameter
- For a binary system with $M = 600M_{\odot}$ and $j = 0$, the ring-down frequency is approximately 19.8 Hz

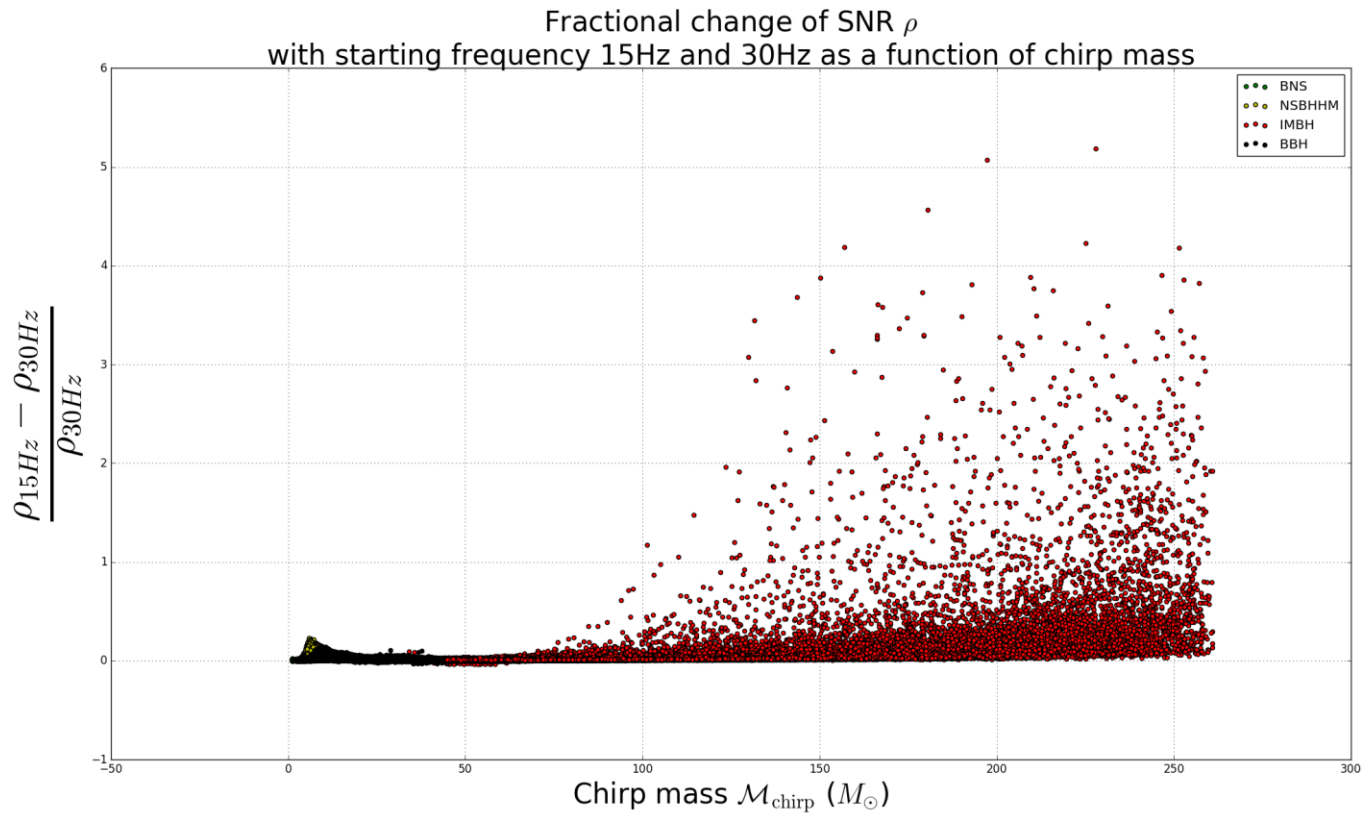
Observing GW Signals from IMBH

- Lower the starting frequency in order to observe them
 - » Uberbank: 30 Hz
 - » IMBH: 15 Hz
- IMBH search was separated from the uberbank search and done off-line
- Challenge: Combining the IMBH bank with the uberbank for **on-line detection**
 - » So we may detect more signals during on-line search
- Q: Should we lower the starting frequency to 15 Hz altogether?
- A: Calculate SNR that compares injections with themselves

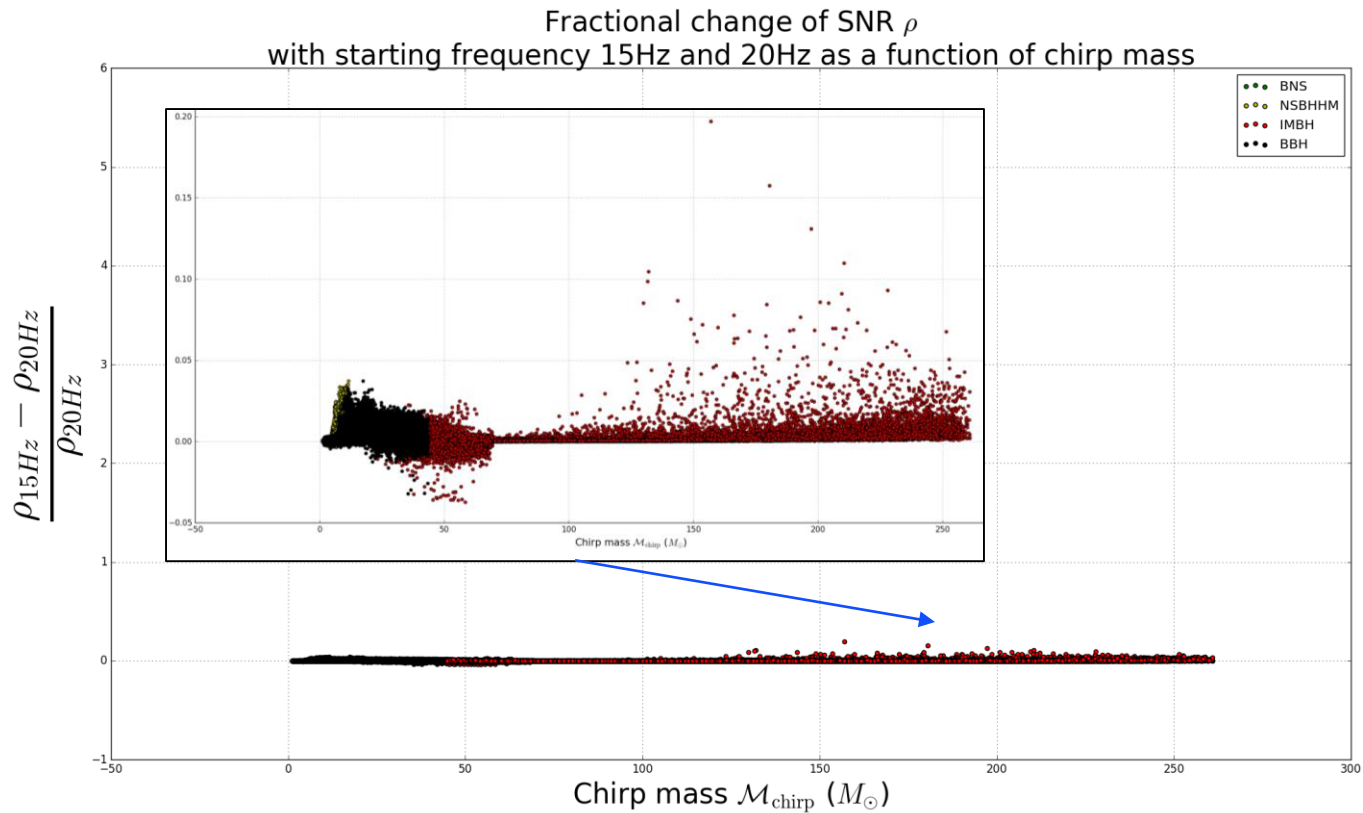
Result: 20 Hz vs. 30 Hz



Result: 15 Hz vs. 30 Hz



Result: 15 Hz vs. 20 Hz



Conclusions from the plots

- A significant increase in SNR starting from $M_{chirp} \sim 60M_{\odot}$
 - » Worth lowering the starting frequency in order to get the extra SNR
 - » Smaller change when further lowering the starting frequency from 20 Hz to 15 Hz
 - » Will not add many computational costs for high mass region since there are fewer templates
- Not significant change in SNR for lower mass region
 - » Lowering the starting frequency in this region will add a lot more costs
 - » But we only get a tiny increase in SNR!
 - » Not worth it!

Future work

- We have modified gstlal codes to accept two different starting frequencies for different regions of chirp mass (within a SVD sub-bank)
- Test our implementation with actual template bank to see the increase in SNR

$h(t)$ Gating

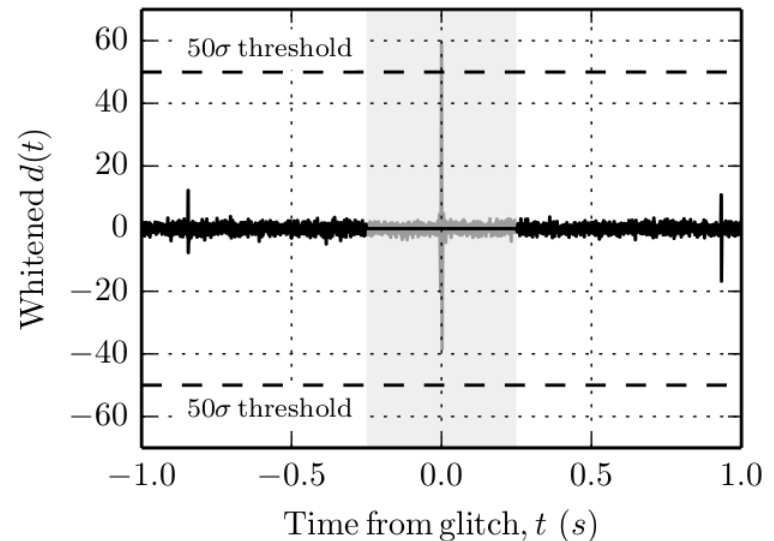
1. Motivation: Why do we need $h(t)$ gating?
2. Problems with the current $h(t)$ gating scheme
3. Linear scale with chirp mass of threshold value

Why do we need $h(t)$ gating?

- Glitches: Short transient instrumental noise fluctuations
- Glitches may cause the filters to have *unreasonably large response*
- For off-line searches: vetoes files for data quality
- Veto seconds of data that contain instrumental glitches
- During on-line mode: no vetoes available

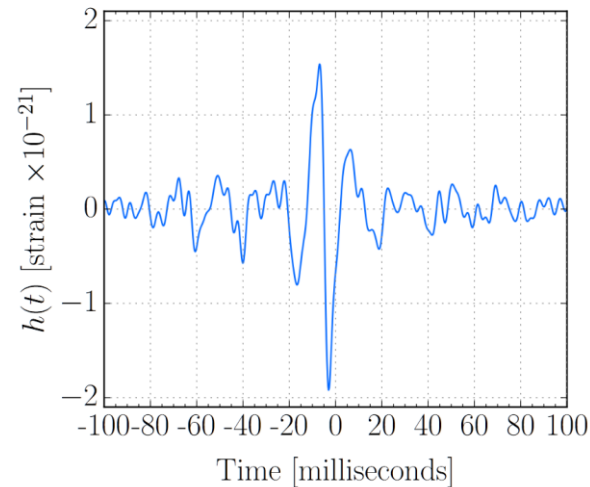
Current $h(t)$ Gating Scheme

- Specify a $h(t)$ gating **threshold value** in advance (before running the search)
- Filter response greater than or equal to the threshold value not likely from the astrophysical sources we are looking for
 - » Currently chosen as 50σ
 - » Lowest value before start missing injections in high mass region
- Replace data with 0's

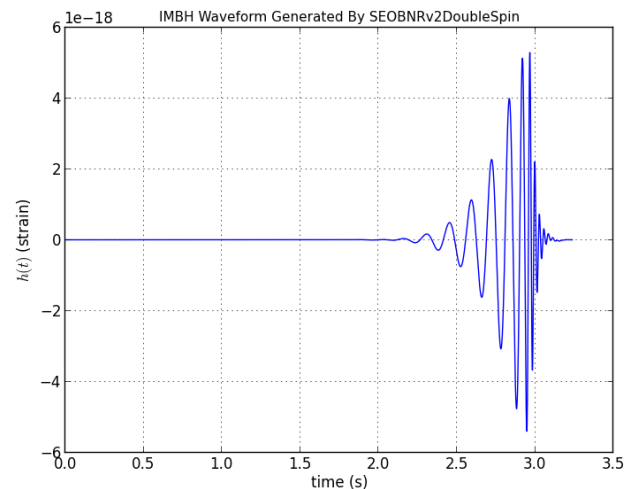


Problems with IMBH GW Signals

- Loud in amplitude
- Short in time
- Easily mimicked by glitches
- Likely to have a higher SNR
- May be 'gated-out' by the current $h(t)$ gating scheme



Blip glitch
Figure taken from [4]



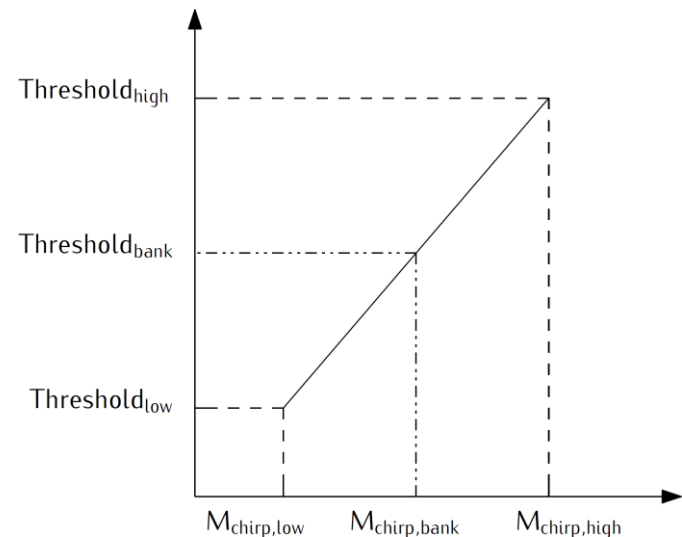
IMBH Waveform

Naïve Solution

- Naively, we can increase the threshold value so that we can detect IMBH signals
- But this will allow more noise in our search
- Also cause more false alarm triggers!
 - » Because we are allowing more glitches, which can mimic actual GW signals easily

Linear Gating Scheme

- Different threshold values for different chirp masses
- Specify a linear scale in advance
- Calculate the ‘appropriate’ threshold value for sub-banks with different masses according to the scale



Testing Linear Gating Scheme

- Ran two gstlal off-line runs:
 1. Default gating (one threshold value for all masses), without vetoes (to simulate the situation of on-line searches)
 2. Linear gating, without vetoes
- Template bank used: uberbank
- Starting frequency: 30 Hz
- Search duration: 3 days
- Linear scale used:

$$M_{\text{chirp, low}} = 0.8; \text{Threshold}_{\text{low}} = 12.0$$

$$M_{\text{chirp, high}} = 45.0; \text{Threshold}_{\text{high}} = 100.0$$

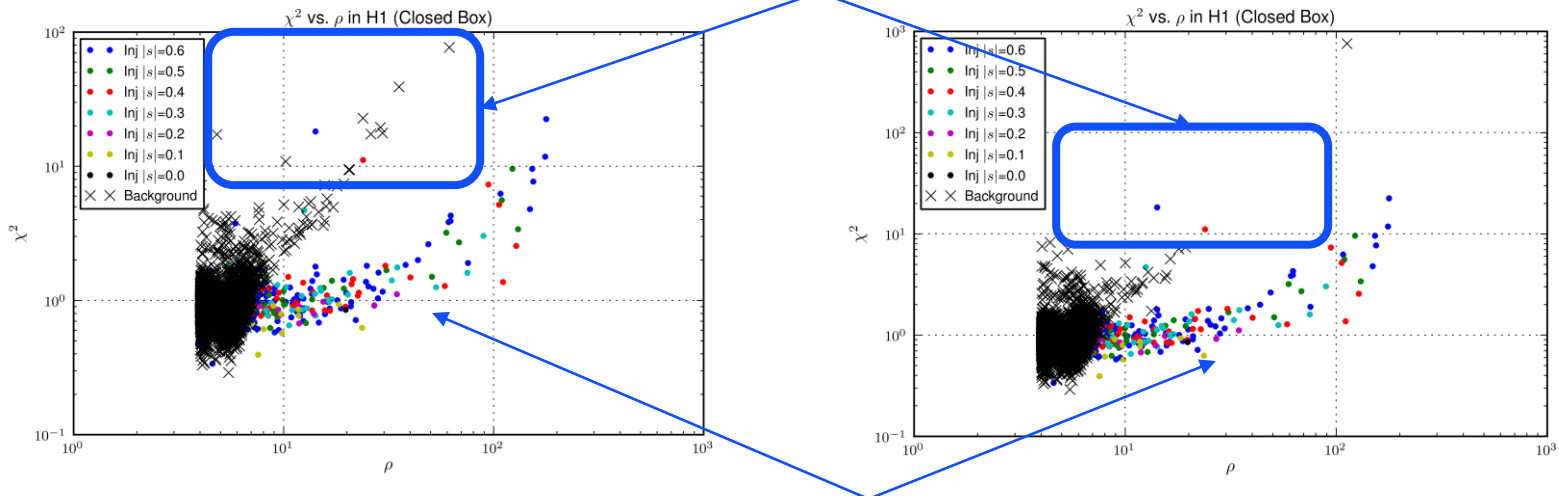
Chirp mass is in the unit of *solar mass*

Results

Default gating, without vetoes

Linear gating, without vetoes

Noise removed with linear gating



All injections recovered in default gating run were also recovered in linear gating run

Conclusions from testing runs

With linear gating scheme, we managed to

- Reduce noise
 - » Clean the data in absence of vetoes
- Recover the same set of injections as compared with the default gating scheme
 - » At least doing as good as the default scheme

Future Work

- Short term:
 - » Exploring different linear scales to get optimal results
 - » Exploring different scales (e.g. Quadratic?)
 - » Re-running with a longer search duration (exposed to more noise)
 - » Re-running with more (software) injections
 - » Re-running with a combined bank (uberbank + IMBH bank)
- Long term:
 - » Applying different threshold values to each template rather than to each sub-bank

Other Projects: Optimizing Bank Splitting Algorithms

- Bank splitting:
 - » Divide the bank into ‘split-banks’
 - » Put 5 split-banks together to form a SVD sub-bank and perform SVD on each sub-bank
- Bank splitting algorithm is
 - » Essential for SVD compression efficiency
 - SVD reduces the no. of effective filters required for the search
 - Finish the search quicker
 - » Essential for Background Estimation
- In progress

Acknowledgement and Reference

Acknowledgement

My sincere gratitude to

- *Mentors:*

- » Surabhi Sachdev
- » Kent Blackburn
- » Alan Weinstein
- » Tjonnie Li @ CUHK GW Group



- *LIGO Laboratory*

- *LIGO SURF Program and Caltech SFP Office*

- *Leonard E Parker Center for Gravitation, Cosmology and Astrophysics @ University of Wisconsin–Milwaukee*

Acknowledgement

This summer research would not be possible without

- National Science Foundation (NSF)
- Charles K. Kao Research Exchange Scholarship
- CN YANG Research Exchange Scholarship
- Department of Physics, CUHK
- M.C. Chu from CUHK

Reference

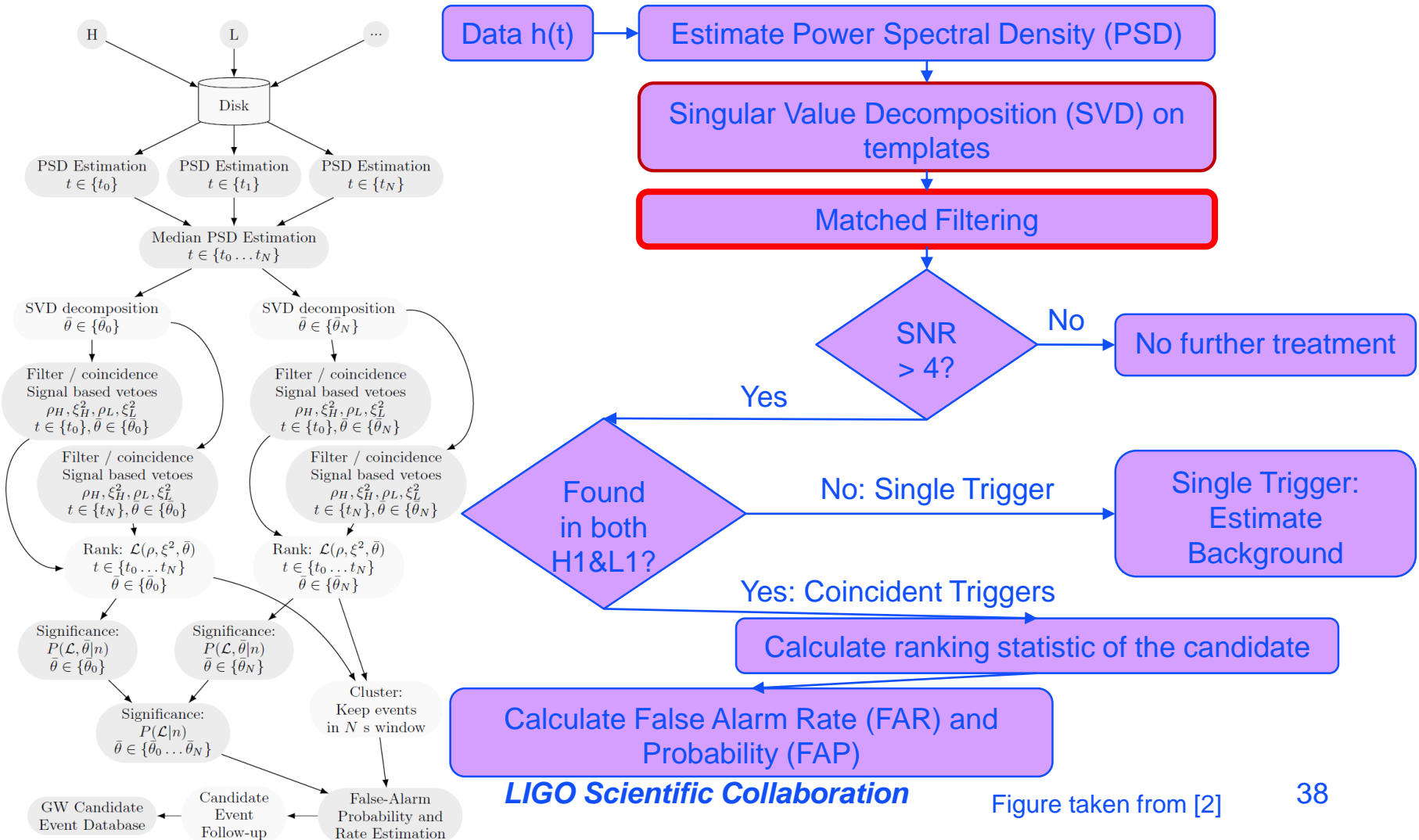
- [1]: The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, and et al. (LIGO Scientific Collaboration and Virgo Collaboration), *Phys. Rev. Lett.* 116, 241103 (2016).
- [2]: C. Messick, K. Blackburn, P. Brady, P. Brockill, K. Cannon, R. Cariou, S. Caudill, S. J. Chamberlin, J. D. E. Creighton, R. Everett, C. Hanna, D. Keppel, R. N. Lang, T. G. F. Li, D. Meacher, A. Nielsen, C. Pankow, S. Privitera, H. Qi, S. Sachdev, L. Sadeghian, L. Singer, E. G. Thomas, L. Wade, M. Wade, A. Weinstein, and K. Wiesner, *ArXiv e-prints* (2016), [arXiv:1604.04324](https://arxiv.org/abs/1604.04324) [astro-ph.IM].
- [3]: The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, and et al., *ArXiv e-prints* (2016), [arXiv:1606.04856](https://arxiv.org/abs/1606.04856) [gr-qc].
- [4]: The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, and et al., *ArXiv e-prints* (2016), [arXiv:1602.03844](https://arxiv.org/abs/1602.03844) [gr-qc].

Q&A

Thank you!

Backup slides

Flow Chart of gstlal



Bank Splitting

1. Motivation: Why do we need to split the template bank?
2. Different ways of splitting the bank
 - Default
 - Sort by chirp time
 - Output rows sort by chirp time

Singular Value Decomposition (SVD)

- One of the techniques adopted by gstlal to speed up the search
- Reduce the number of ‘effective’ filters required for the search
- Decompose any matrix $A_{m \times n}$ into three matrices

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T$$

- Where U, V are orthogonal matrices forming the singular vectors and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{r=\text{rank}(A)})$
- $\sigma_1, \sigma_2, \dots, \sigma_r \geq 0$ are called singular values and are ordered in descending order

Lower Rank Matrix Approximation (LRMA) using SVD

- We can express the matrix as a summation of singular values and the singular vectors

$$A = \sum_{i=1}^r \sigma_i \vec{u}_i \vec{v}_i^T$$

- To approximate A with a lower-rank matrix M (where $\text{rank}(M) = k$), the **best** you can do is to use

$$M = \sum_{i=1}^k \sigma_i \vec{u}_i \vec{v}_i^T \quad \text{with} \quad \min \|A - M\|_F$$

- Meaning that the best LRMA is *truncating* the summation

Performing SVD on Templates

- A template is essentially a time series of **numbers**
 - » Sampling the ‘simulated signal’ at certain frequency
 - » Can be put in the form of matrix easily
- When performing SVD, we put many (~1000) templates together in a matrix

Most SVD algorithms
require that $m \geq n$

$$A = \begin{pmatrix} 0.012 & \cdots & 0.065 \\ \vdots & \ddots & \vdots \\ -0.021 & \cdots & 0.033 \end{pmatrix} \downarrow$$

Each column is
a template

- Therefore m is the number of samples in each template and n is the number of templates in a SVD sub-bank

Splitting Template Bank

- As a result, the way we put templates together in a sub-bank is crucial
 - » Affect the compression effect of SVD and hence the performance of the pipeline
 - » Affect the sensitivity of the search since matched filtering jobs are done for each SVD sub-bank for each reference PSD
- **Default bank splitter:** first bin templates by χ_{eff} (effective spin parameter) then sorting templates by chirp mass within each bin
 - » Trying to put similar waveforms together in a split bank
- Group a number of templates in a split-bank
- Each SVD bank consists of 5 split-banks

Template Bank

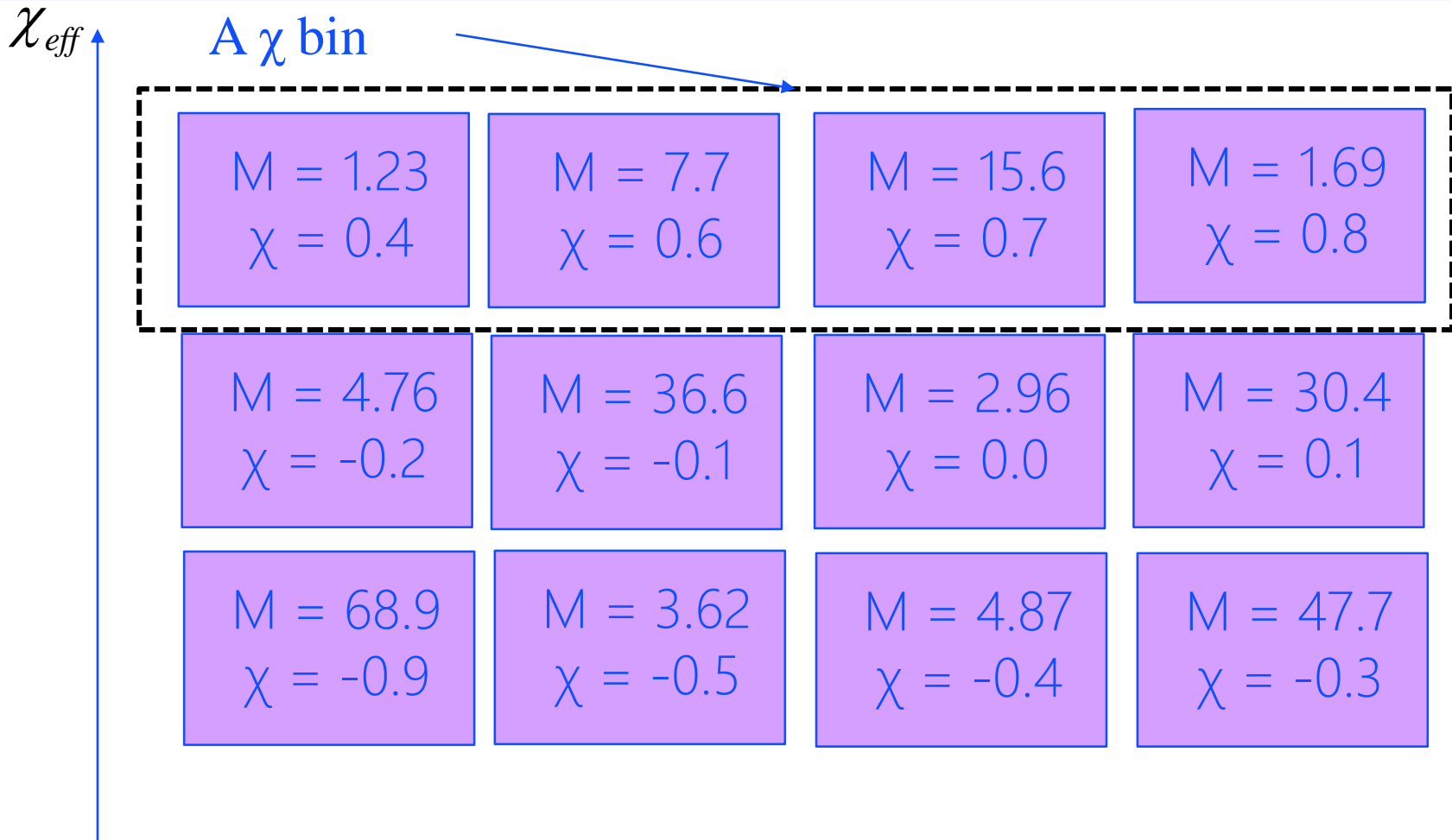
$M = 1.23$ $\chi = 0.4$	$M = 3.62$ $\chi = -0.5$	$M = 15.6$ $\chi = 0.7$	$M = 30.4$ $\chi = 0.1$
$M = 47.7$ $\chi = -0.3$	$M = 4.76$ $\chi = -0.2$	$M = 36.6$ $\chi = -0.1$	$M = 2.96$ $\chi = 0.0$
$M = 7.7$ $\chi = 0.6$	$M = 68.9$ $\chi = -0.9$	$M = 4.87$ $\chi = -0.4$	$M = 1.69$ $\chi = 0.8$

M stands for **chirp mass** with the unit of *solar mass*
 X stands for the dimensionless effective spin parameter

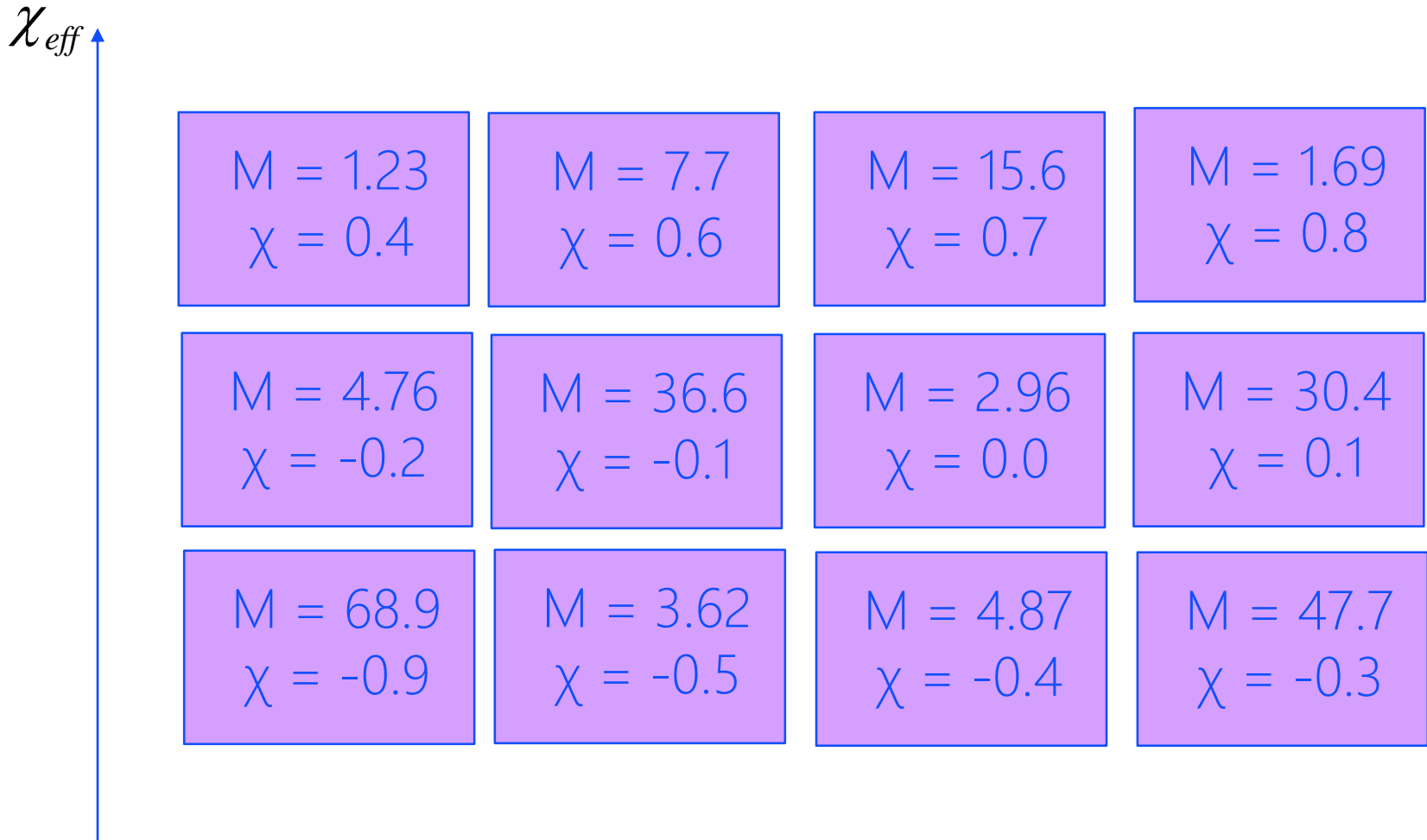
Step 1: Binning By χ_{eff}

$M = 1.23$ $\chi = 0.4$	$M = 3.62$ $\chi = -0.5$	$M = 15.6$ $\chi = 0.7$	$M = 30.4$ $\chi = 0.1$
$M = 47.7$ $\chi = -0.3$	$M = 4.76$ $\chi = -0.2$	$M = 36.6$ $\chi = -0.1$	$M = 2.96$ $\chi = 0.0$
$M = 7.7$ $\chi = 0.6$	$M = 68.9$ $\chi = -0.9$	$M = 4.87$ $\chi = -0.4$	$M = 1.69$ $\chi = 0.8$

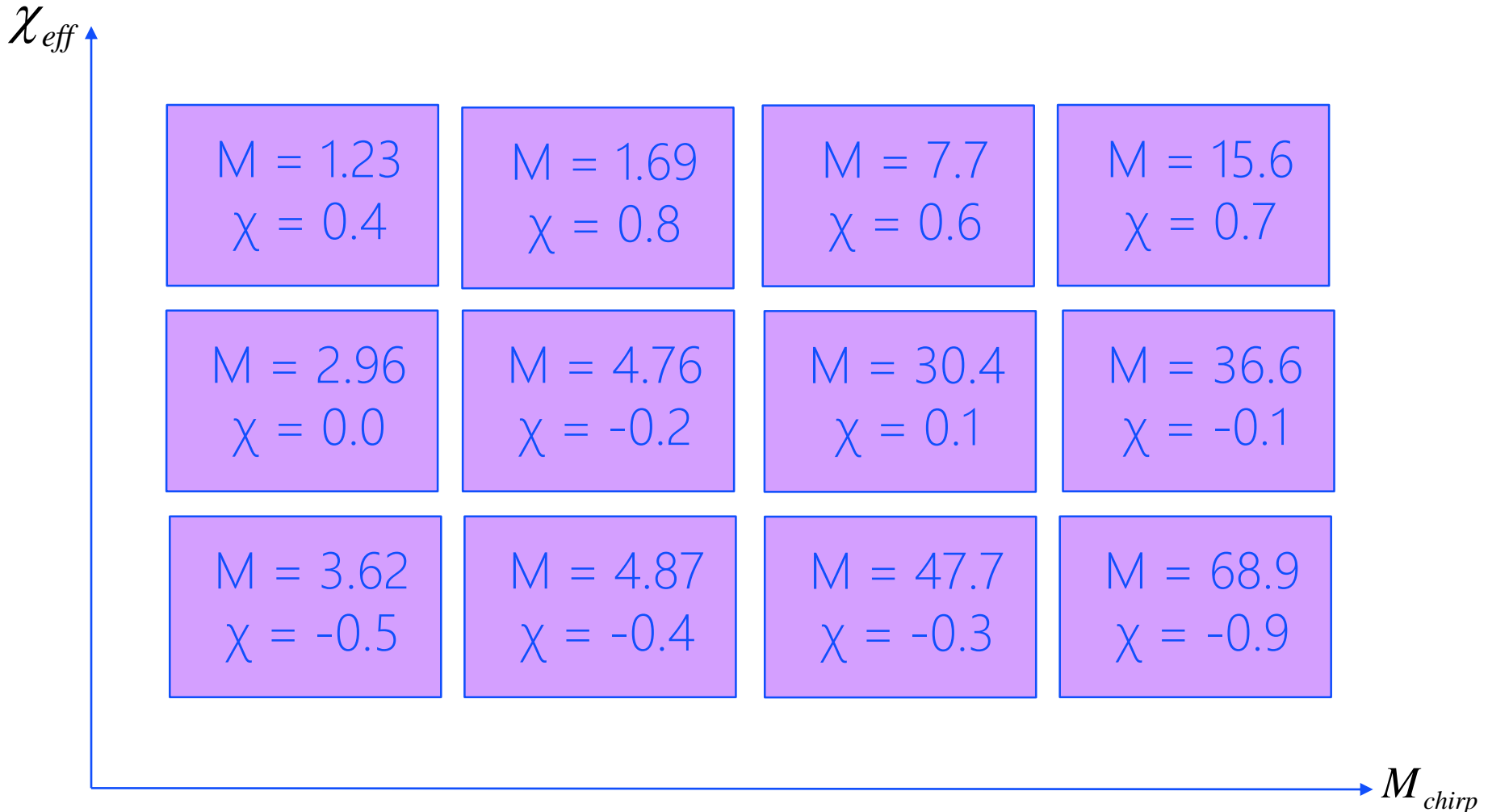
Step 1: Binning By χ_{eff}



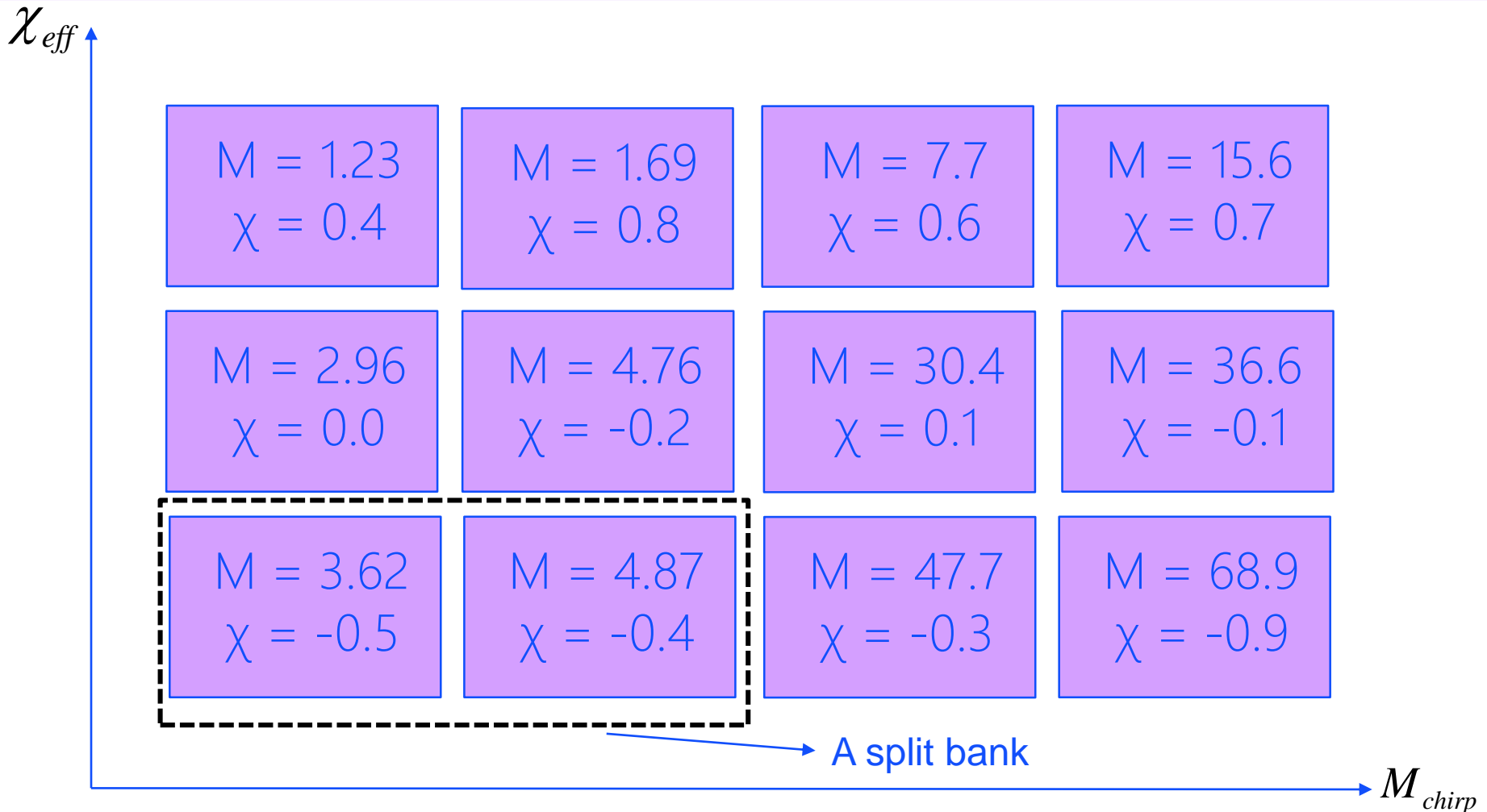
Step 2: Sorting By M_{chirp}



Step 2: Sorting By M_{chirp}



Step 2: Sorting By M_{chirp}



Output of bank splitter: split-banks

0	$M = 1.23$ $\chi = 0.4$	$M = 1.69$ $\chi = 0.8$	$M = 7.7$ $\chi = 0.6$	$M = 15.6$ $\chi = 0.7$	3
1	$M = 2.96$ $\chi = 0.0$	$M = 4.76$ $\chi = -0.2$	$M = 30.4$ $\chi = 0.1$	$M = 36.6$ $\chi = -0.1$	4
2	$M = 3.62$ $\chi = -0.5$	$M = 4.87$ $\chi = -0.4$	$M = 47.7$ $\chi = -0.3$	$M = 68.9$ $\chi = -0.9$	5

New Proposals for Bank Splitter

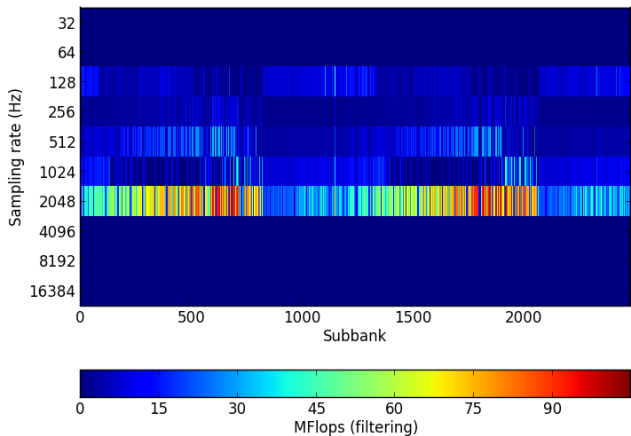
- **Proposal I: Sort by chirp time only**
 - » We expect that the distribution of the time required to complete the matched filtering will be *more dispersed*
- **Proposal II: Sub-banks sort by the chirp time of their first template**
 - » Since each SVD bank comprises of 5 split banks, we are sort of putting similar templates together
- Chirp time of a template is estimated by *SEOBNRv2DoubleSpinChirpTime* function implemented in lalsimulation

Comparing bank splitters

- There are mainly two criteria:
 1. Speed of matched filtering jobs
 2. Sensitivity of the search
- We can compare bank splitters by looking at
 1. For speed:
 1. Flopulator (FLOP Calculator)
 - Calculate the number of Floating Point Operations needed for the filtering jobs WITHOUT actually doing them
 2. `gstlal_inspiral` time
 - Measuring the actual time required to finish the filtering jobs
 2. For sensitivity:
 1. Range
 - How far can we detect a GW signal

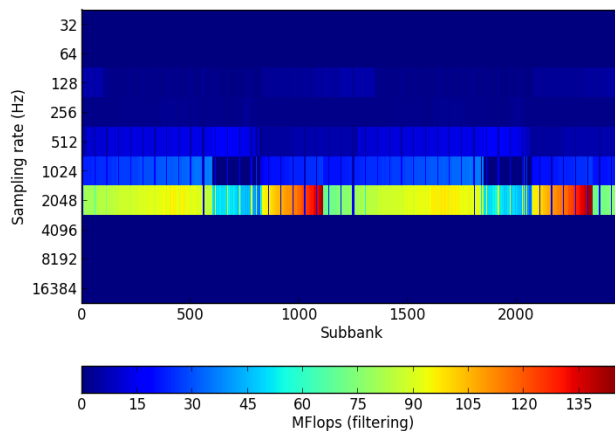
Results: Flopulator output

Total MFlops (filtering): 183614.44



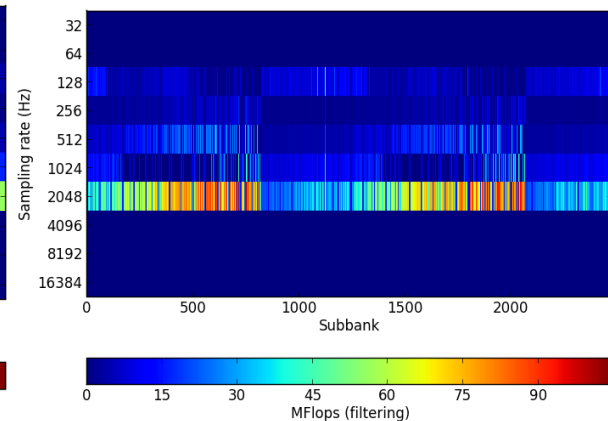
Default

Total MFlops (filtering): 295843.40



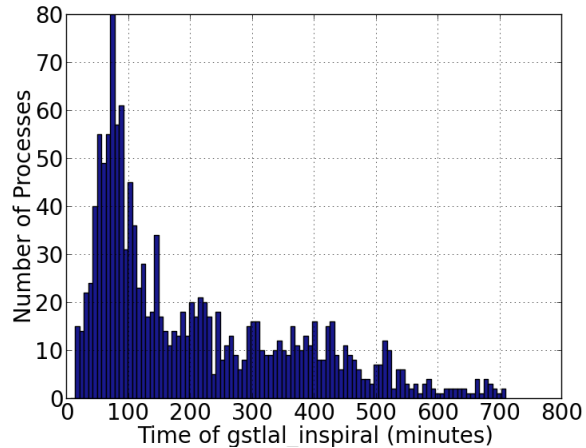
Proposal I

Total MFlops (filtering): 183614.44

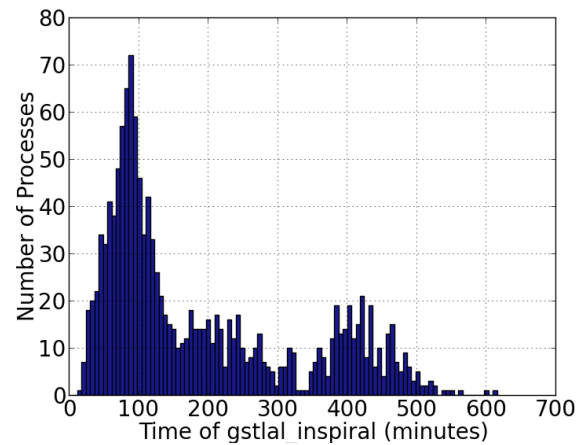


Proposal II

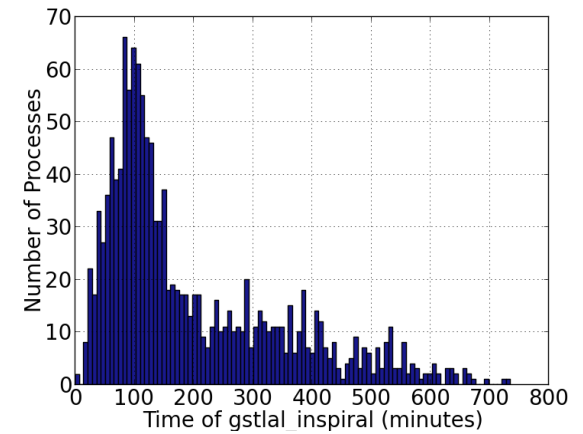
Results: gstlal_inspiral Time



Default

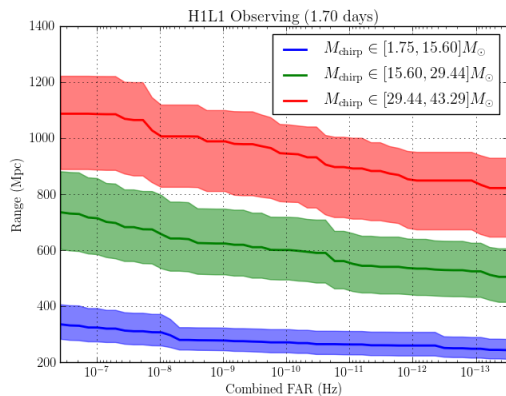


Proposal I

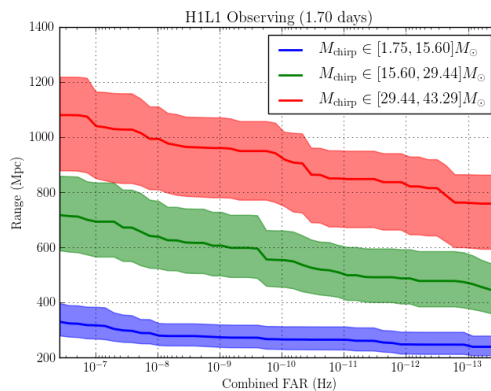


Proposal II

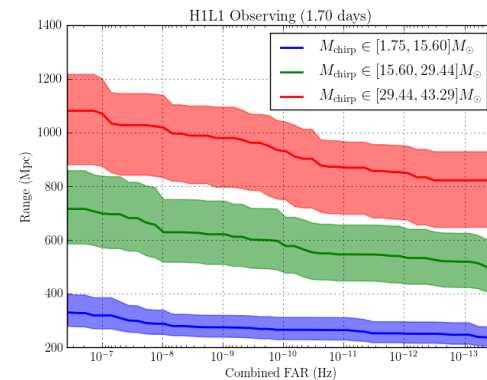
Results: Range



Default



Proposal I



Proposal II

Quick Summary on Bank Splitting

- Put templates with similar chirp time together and form a split bank for SVD
 - » Chirp time is a good criterion about how templates match with each other for Inspiral-Merger-Ringdown region
- As compared with the default bank splitter, chirp time bank splitter:
 - » The greatest time required to complete matched filtering (in `gstlal_inspiral`) is lower
 - » Higher total FLOPS required for matched filtering predicted by Flopulator (FLOP calculator)
 - » Same sensitivity (within the error)

Quick Summary on Bank Splitting

- There is another group working on bank splitting as well (Leslie Wade @ Kenyon College)
- Because of a bug, they group templates randomly instead of chirp time
- A better sensitivity is observed
- This raises two concerns
 1. The current standard of comparing binning methods is flawed
 2. Current ranking statistic is flawed because we have been assuming that grouping unlike templates will only decrease the sensitivity