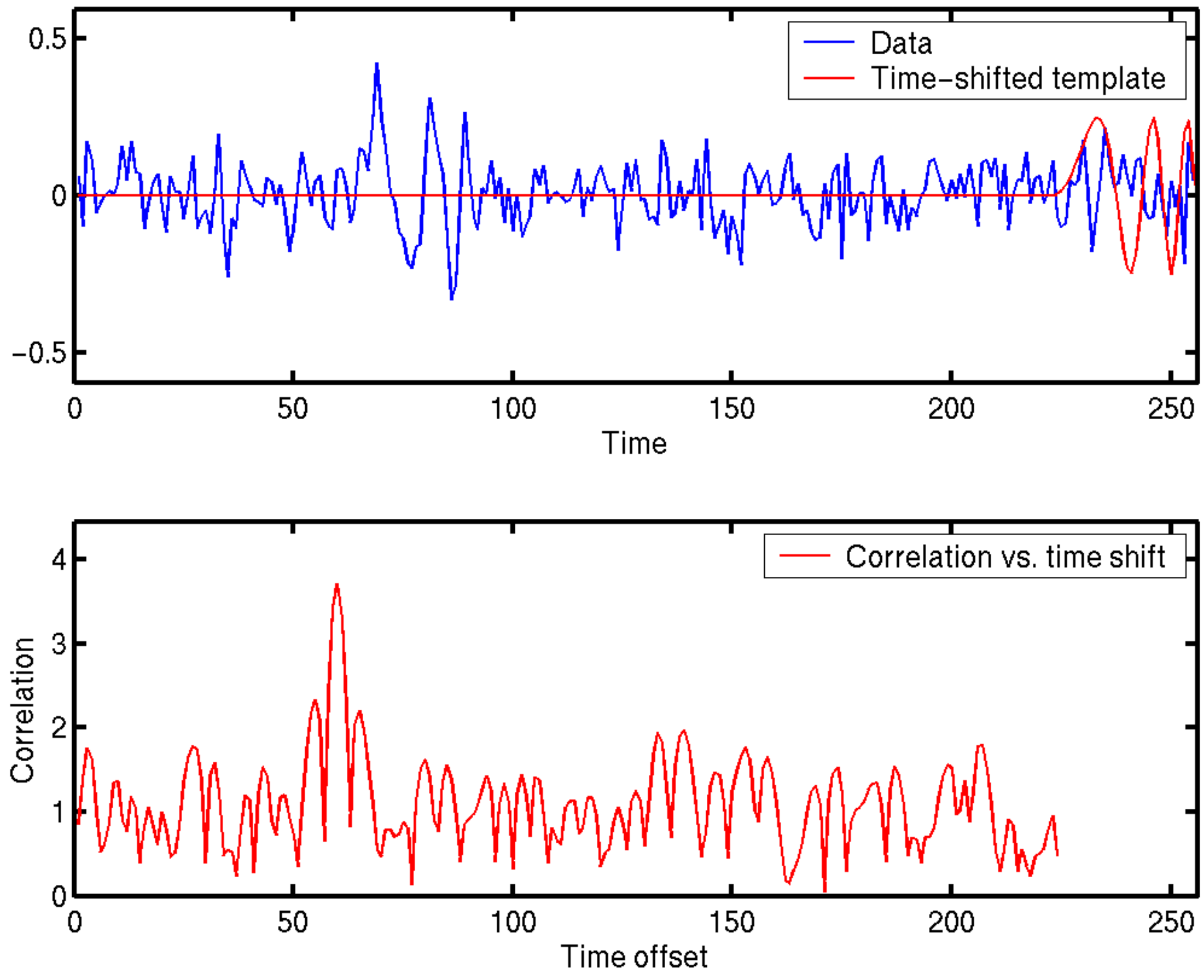
Matched Filtering in GW Data Analysis

Peter Shawhan

Fermi Summer School
June 4, 2018

Basic Illustration of Matched Filtering

<u>Correlate</u> data with expected signal (Here, plotting absolute value)



Other Ways to Think About Matched Filtering (1)

Correlating data with template can be thought of as taking an *inner product*

$$C = \langle s \,|\, h \rangle = \int_{-\infty}^{\infty} dt' \, s(t') \, h(t') \quad or \quad \sum_j s_j h_j$$

Data — Template

For GW data analysis, there is usually a free param: time offset

$$C(t) = \int_{-\infty}^{\infty} dt' \, s(t') \, h(t' - t) \qquad or \qquad C_i = \sum_j s_j h_{j-i}$$

Time offset — Data — Template with time offset

Yields a time series of correlation values

Other Ways to Think About Matched Filtering (2)

Correlating data with template is equivalent to an *FIR filter* with coefficients following the template

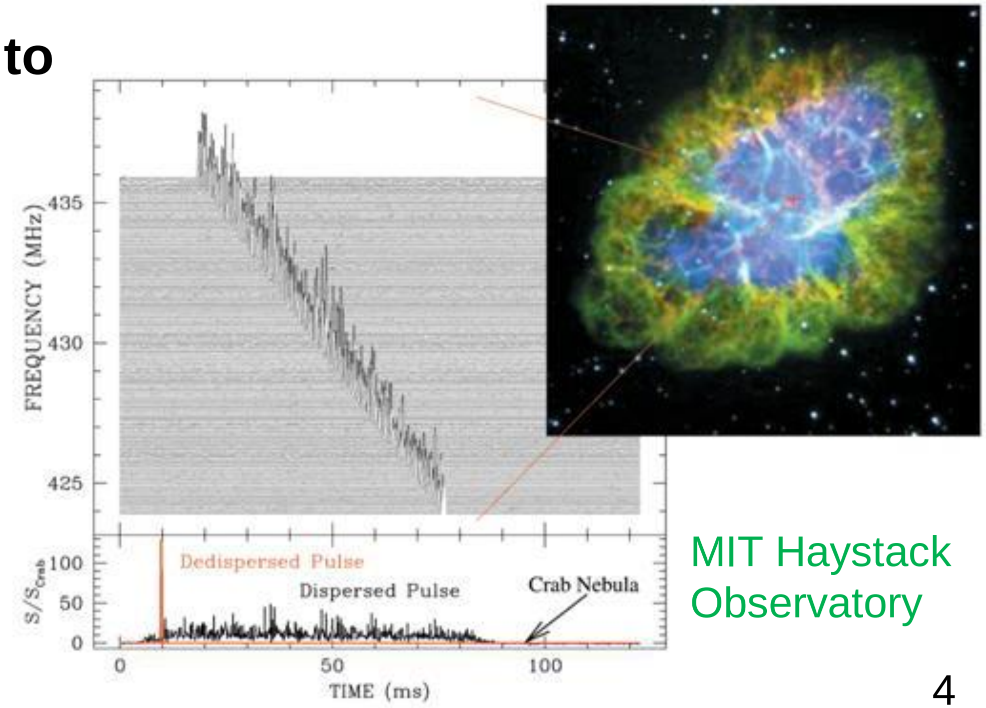
$$y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + \Lambda + b_N x_{n-N}$$

The impulse response of that FIR filter looks like the template, but *time-reversed*

The goal of this kind of filter is to "compress" an extended signal into a delta function

Shift all parts of the signal in the data to a common time, and add them together with the same sign

Similar to "de-dispersion" in radio telescope pulse search

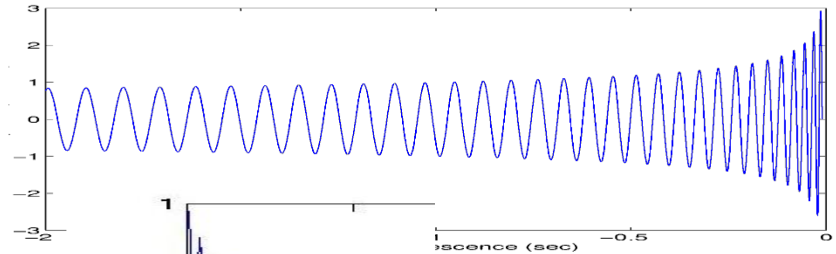
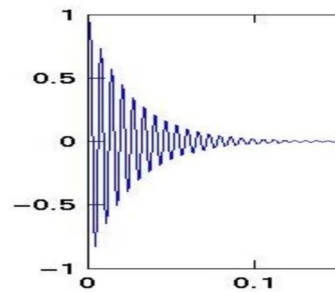
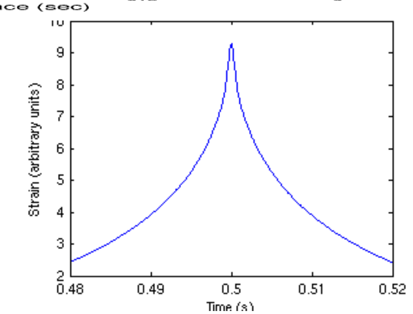


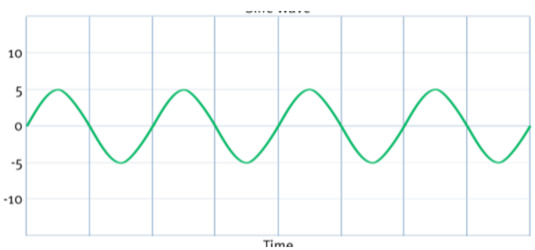
MIT Haystack Observatory

Usefulness

Useful for GW data analysis whenever target waveform is known

- Binary coalescence
- Ringdown
- Cosmic string cusp
- Continuous-wave signal



Phase coherence is more important than amplitude matching

Also known as "Wiener optimal filter"

- Optimal detection statistic if noise is Gaussian

Books:
Creighton & Anderson,
"GW Physics and Astronomy";
Wainstein and Zubakov,
"Extraction of Signals from Noise"

Matched Filtering in Frequency Domain

$$C(t) = \int_{-\infty}^{\infty} dt' \, s(t') \, h(t'-t)$$

Time offset

Data

Template with time offset

Rewrite correlation integral using Fourier transforms…

$$\Rightarrow \quad C(t) = 4 \int_{0}^{\infty} \tilde{s}(f) \, \tilde{h}^{*}(f) \, e^{2\pi i f t} \, df$$

This is simply the inverse Fourier transform of $\tilde{s}(f) \, \tilde{h}^{*}(f)$ **!**

(Correlation in time domain is product, with complex conj, in freq domain)

A Fast Fourier Transform (FFT) is a computationally efficient way to calculate filter output for <u>all</u> time offsets!

Optimal Matched Filtering with Frequency Weighting

FFT of data

Template can maybe be generated in frequency domain using stationary phase approximation

$$C(t) = 4 \int_0^\infty \frac{\tilde{s}(f)\ \tilde{h}^*(f)}{S_n(f)}\ e^{2\pi i f t}\ df$$

Noise power spectral density

This de-weights frequencies with larger noise power

Or, can apply a time-domain filter which is essentially the inverse Fourier transform of $\tilde{h}(f)/S_n(f)$

Look for maximum of $|C(t)|$ **above some threshold ➔ trigger**

See, for instance, Allen et al., PRD 85, 122006 (2012)

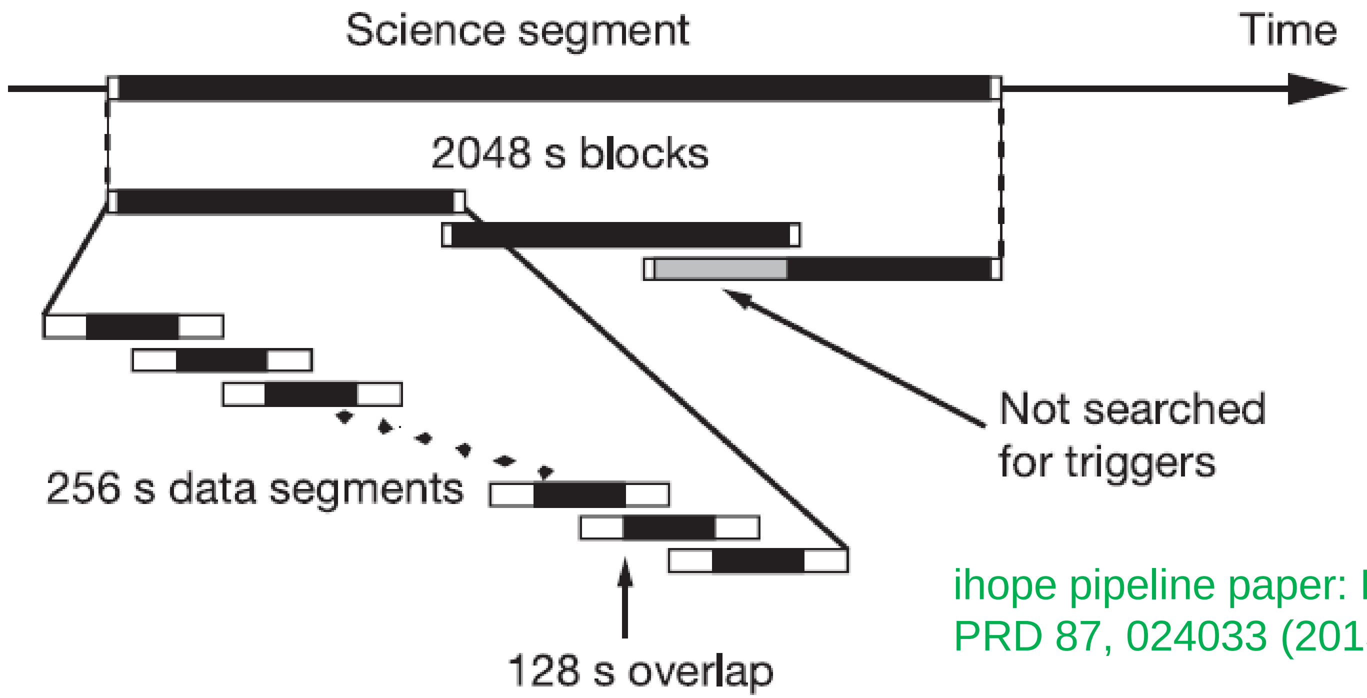
Searching a Full Data Set

Search *overlapping* intervals to avoid wrap-around effects when filtering in frequency domain

Do inverse FFTs on, say, 256 s of data at a time

Estimate power spectrum from longer stretches of data, e.g. using median

Original iLIGO scheme: D. A. Brown for the LSC, CQG 22, S1097 (2005)



ihope pipeline paper: Babak et al., PRD 87, 024033 (2013)

Binary Coalescence Source Parameters vs. Signal Parameters

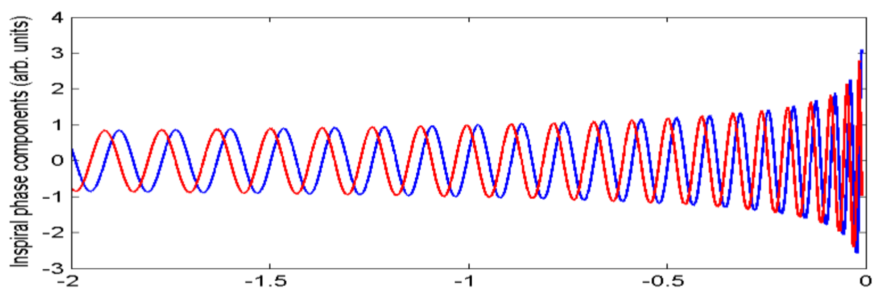
Inspiral source parameters

- Masses (m1, m2)
- Spins → Nearly degenerate with mass ratio
- Orbital phase at coalescence → Maximize analytically when filtering
- Inclination of orbital plane
- Sky location

→ Simply multiplicative for a given detector (if no precession, and in long-wavelength limit)

- Distance → Simply multiplicative
- Coalescence time

Filter with orthogonal templates, take quadrature sum



➔ Only have to explicitly search over masses and coalescence time (“intrinsic parameters”)

Template Matching

Want to be able to detect any signal in a *space* of possible signals

All with different phase evolution

… but do it with a finite set of templates! i.e., a "template bank"

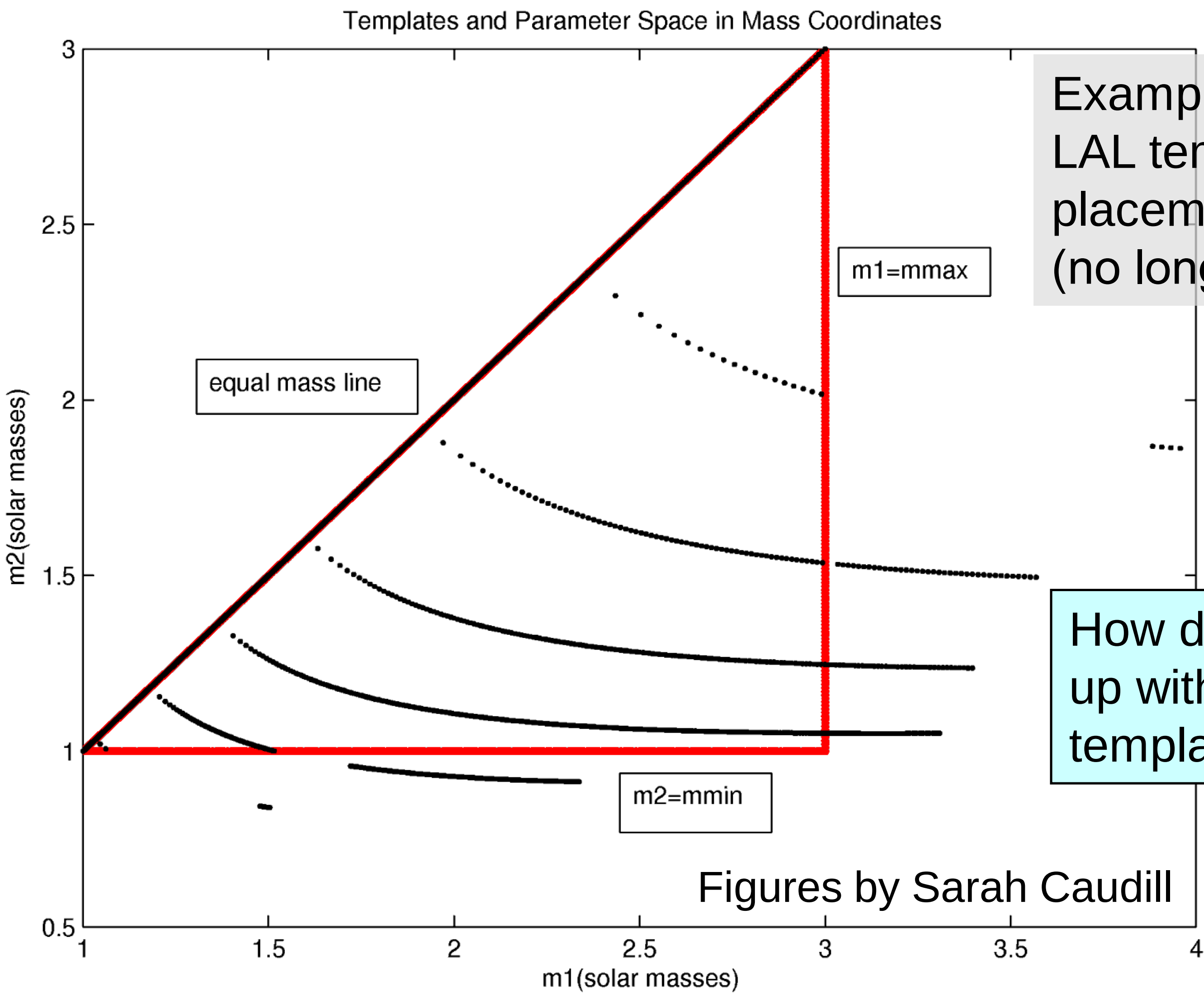
So make sure there is a "close enough" template for every part of the signal space

Require a minimum overlap between signal and template, e.g. 0.97

Often can calculate a "metric" which parametrizes the mismatch for small mismatches

See, for instance, Sathyaprakash and Dhurandhar, PRD 44, 3819 (1991);
Balasubramanian, Sathyaprakash and Dhurandhar, PRD 53, 3033;
Owen and Sathyaprakash, PRD 60, 022002 (1999)

Template Bank Construction

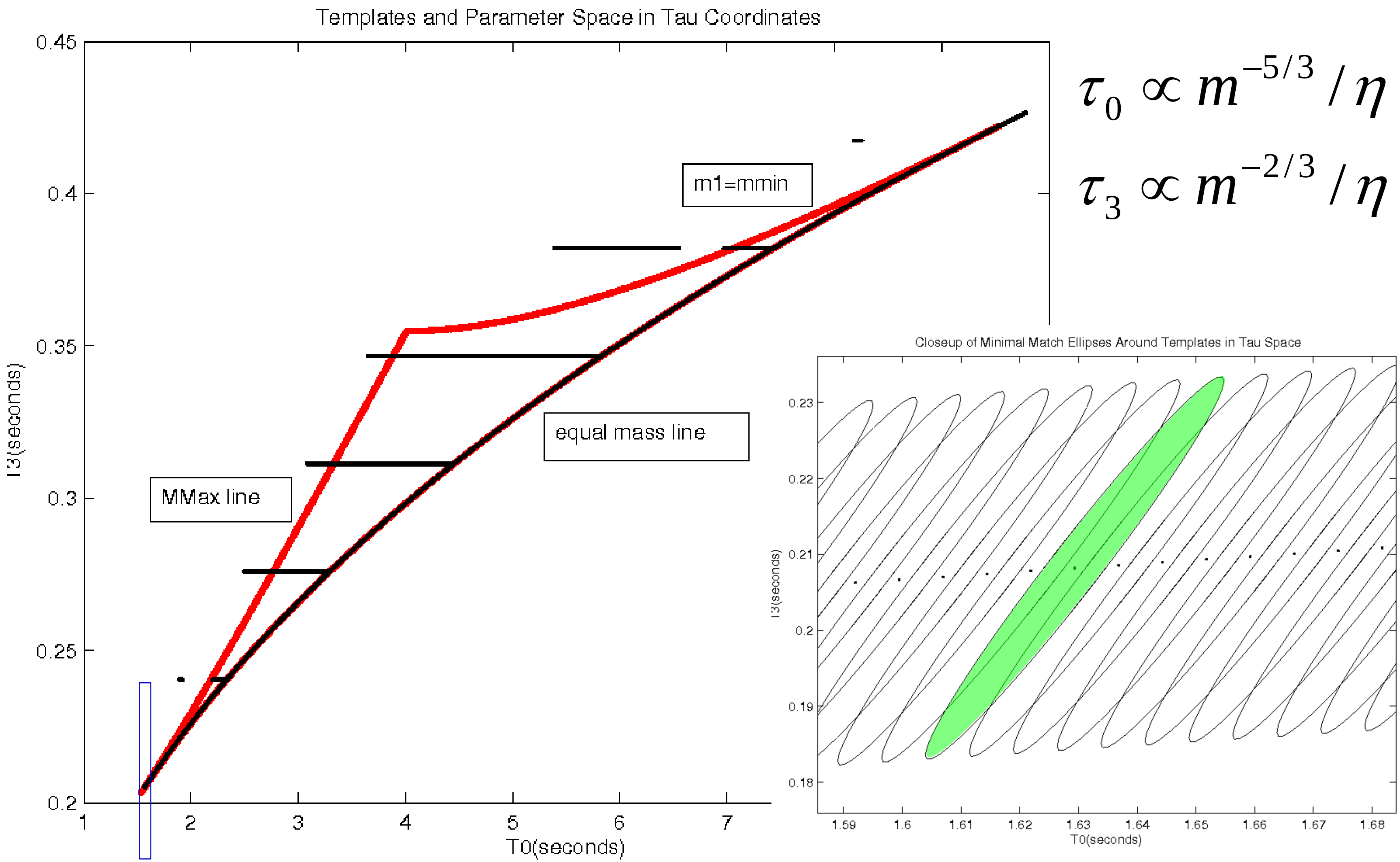


Example from early LAL template bank placement algorithm (no longer in use…)

How did we come up with this set of templates???

Figures by Sarah Caudill

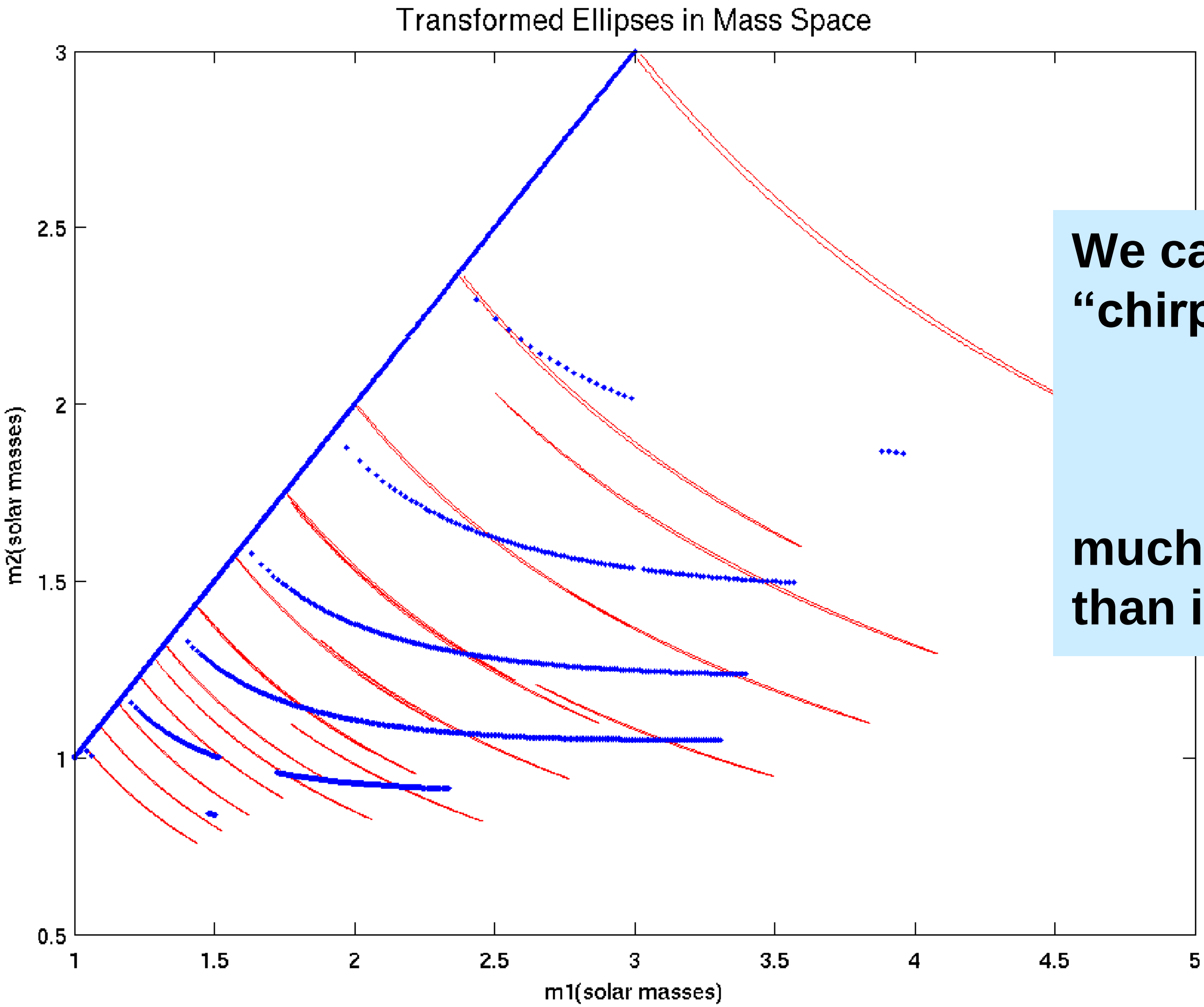
Template Bank Construction in (τ_0,τ_3) space



$$\tau_0 \propto m^{-5/3} / \eta$$

$$\tau_3 \propto m^{-2/3} / \eta$$

Ellipses in Mass Space



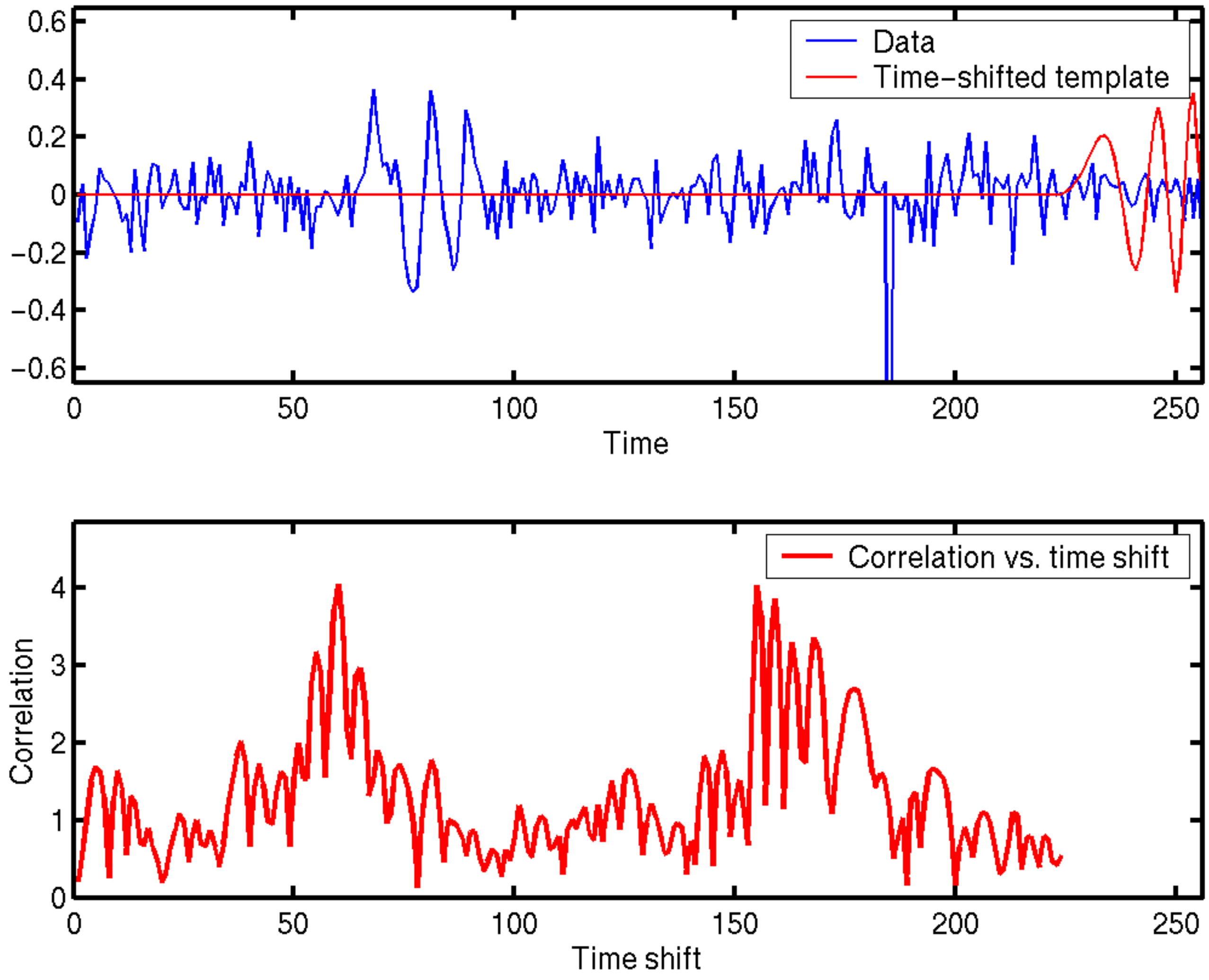
We can determine the "chirp mass",

$$\frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$$

much more precisely than individual masses

Peters and Mathews, Phys. Rev. 131, 435 (1963)

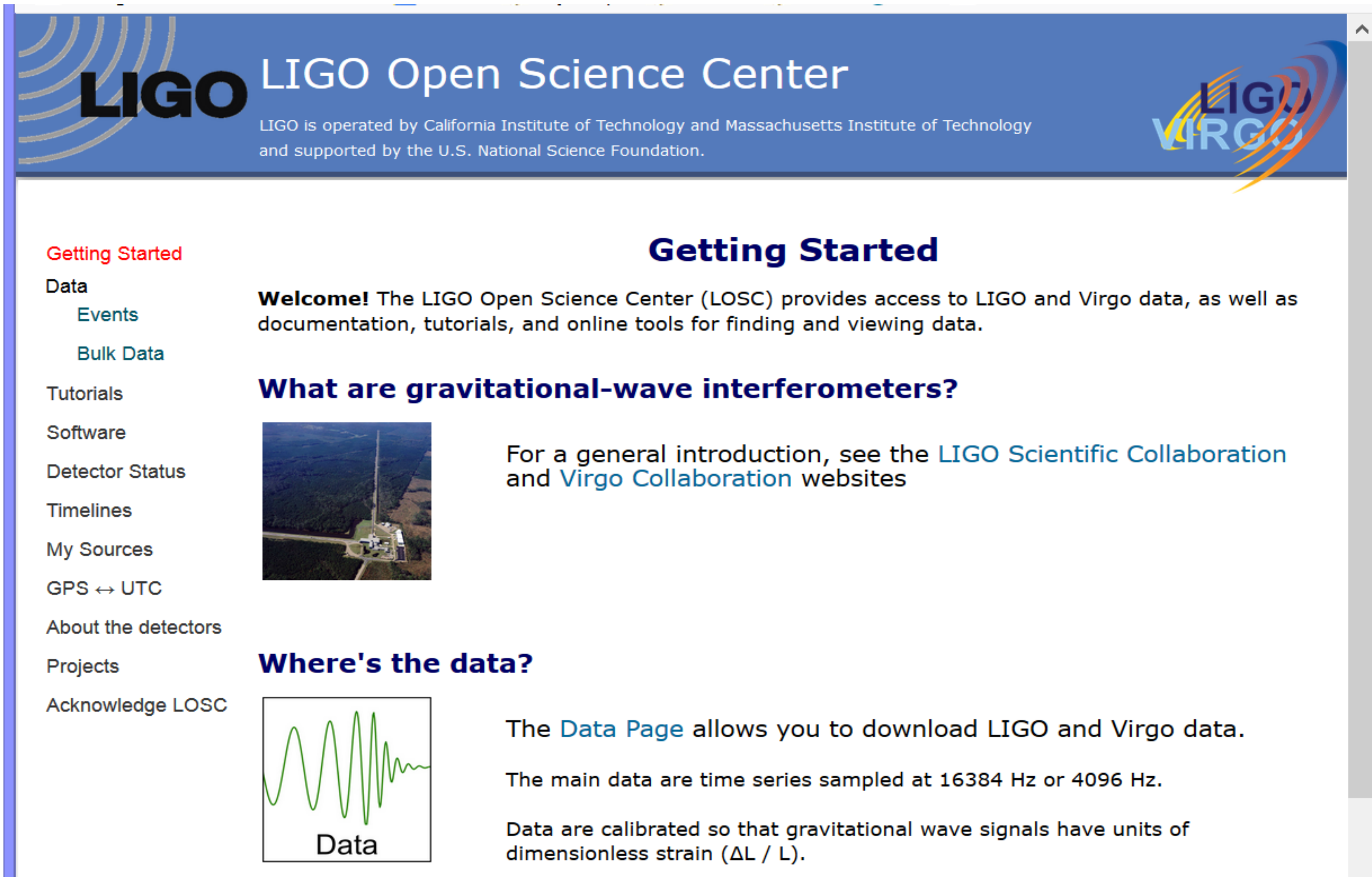
Matched Filtering is *Sensitive*, But Not *Selective*



Working with LIGO (and Virgo) Data

See the LIGO Open Science Center, http://losc.ligo.org

(Will eventually be renamed the Gravitational Wave Open Science Center)



Tutorials on LOSC web site

GPS ↔ UTC
About the detectors
Projects
Acknowledge LOSC

Open Data Workshop Web Course (2018)

Self-paced web course on LIGO data analysis

Course Material

newest

Binary Black Hole Events

Use matched filtering to find signals hidden in noise.

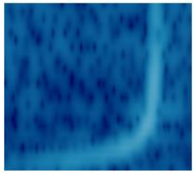
Run: Azure | mybinder (Beta)

View: GW150914 | LVT151012 | GW151226 | GW170104

Download: zip file with data | Jupyter notebook | python script

Can run in regular python environment (native or in docker), or use Azure or binder web-based engines

Quickview Notebook



Make summary plots for any short segment of LIGO data.

Run: Azure | mybinder (Beta)

Download: IPython 4

Introduction to LIGO Data Files
[**Run:** workspace]

- Step 0) Software Setup
- Step 1) Download LIGO Data
- Step 2) What's in a LIGO Data File?
- Step 3) Working with Data Quality
- Step 4) Using the example API (readligo)

oldest

Working with Data
[**Run:** workspace]

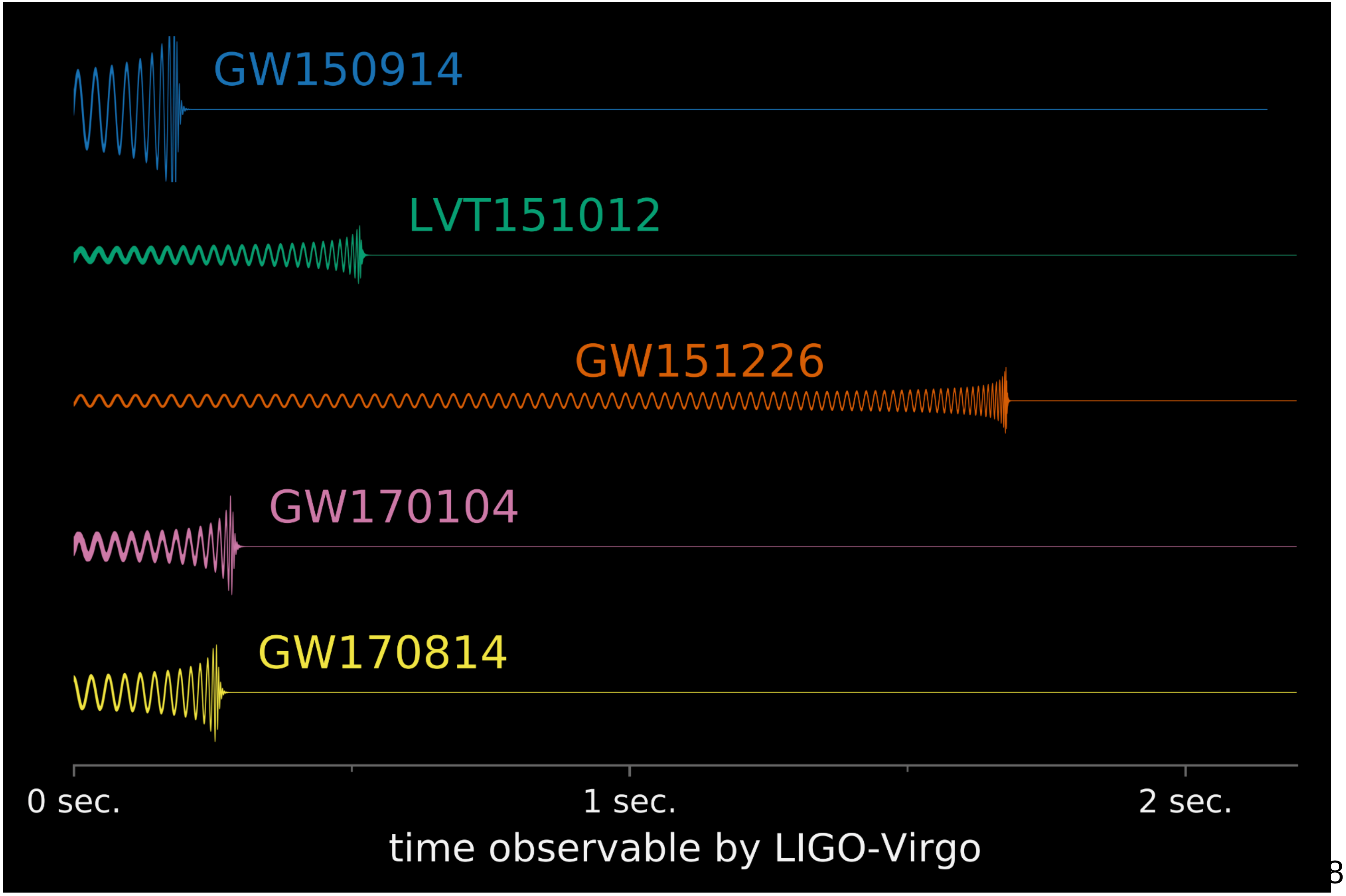
- LOSC Example API
- Working with Segment Lists
- FFTs, PSDs, and Spectrograms:
  - Lots of Plots tutorial
  - Browse the plot gallery

Suggested Exercises for Today - 1

Run the Quickview tutorial in Azure (Jupyter notebook)

1. In the Quickview Notebook section of the page, click on the ‘Azure’ link
2. You may be able to log in with your institutional email address.
   Or log in with an existing Microsoft account, or create one for free.
3. Clone the library (using button near middle of page)
4. Click on quickview.ipynb to open the notebook
5. It opens showing the 2010 “Big Dog” hardware injection (fake signal)
6. In the second cell, change the dataset to ‘O1’ and the target time (t0) to 1126259462.4 (approximately the coalescence time of GW150914)
7. Do Cell → Run All
   It takes a minute or so for the script to run and the figures to appear
8. Study the outputs – e.g. note use of whitening and band-pass filtering
9. Now display the data around the time of GW151226 (t0=1135136350.65); adjust the time axis to show +/- a few seconds.
   Do you believe that there is an event there in the data?

Event durations



Suggested Exercises for Today - 2

Use matched filtering to find GW150914

1. From the Azure library listing, click LOSC_Event_tutorial.ipynb to open that notebook
2. Run the notebook
3. Check out all the plots
4. Listen to the audio files for the template and for the whitened data, for H1 and also for L1

Use matched filtering to find GW151226

1. In the second cell of the notebook, uncomment the GW151226 line
2. Run the notebook
3. Check out all the plots… note the much longer duration of the signal in the template plots, and make sure you understand the matched filter output plots
4. Can you hear the signal in the audio files?

If you have more time – option 1

Implement matched filtering from scratch in your preferred computing environment

- It's pretty straightforward to do basic time-domain matched filtering
- Get an H1 strain data file from around the time of GW150914, either from http://losc.ligo.org/events/ or from the bulk data archive
- Use this for the template: https://losc.ligo.org/s/events/GW150914/P150914/fig2-unfiltered-waveform-H.txt
- When you have that working, try to figure out Fourier transforms and do the matched filtering in the frequency domain
- Now take the noise spectrum into account, i.e. do *optimal* matched filtering. The amplitude spectral density can be obtained from http://losc.ligo.org/events/GW150914/ ; square it to get the power spectral density, $S_n(f)$.

Matlab example: see Wednesday Matlab exercises and solutions at http://tinyurl.com/GWAdata

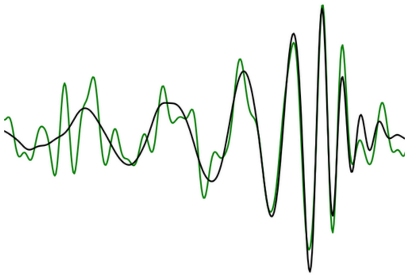
If you have more time – option 2

Check out the Open Data Workshop Web Course

Open Data Workshop Web Course (2018)

Self-paced web course on LIGO data analysis

Course Material



LIGO Scientific Collaboration

Open Data Workshop #1

Sunday - Tuesday, March 25 - 27, 2018

Data Workshop | Location | Lodging | Transportation | Registration | Program | Lecture Videos

Workshop Web Course

Overview

These are materials from the 2018 LSC Open Data Workshop. The web course:

- includes 5 hours of lecture
- includes 10-30 hours of data analysis programming excercises
- is intended for people holding or pursuing a graduate degree in physics, astronomy, or a related field
- targets learning objectives related to gravitational wave data analysis using LIGO and Virgo

Prerequisites

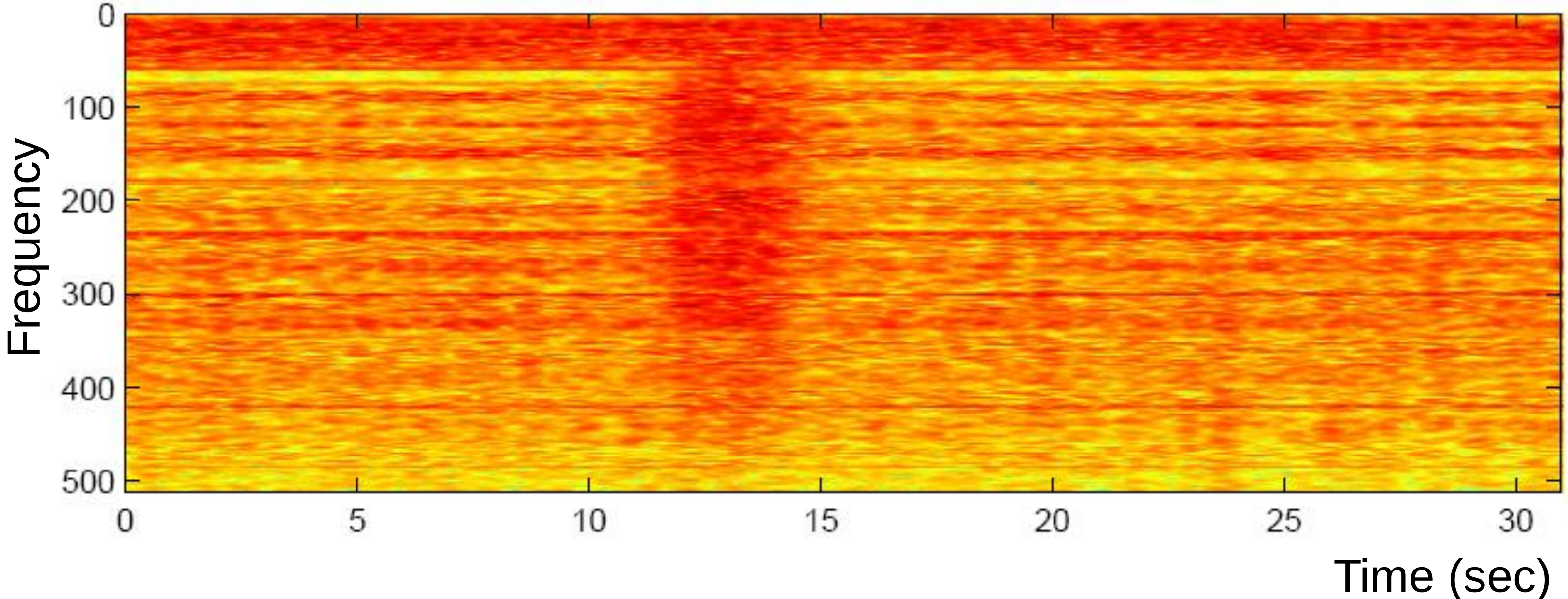
Introduction to LIGO and Virgo

If you are not familiar with LIGO and Virgo, see:

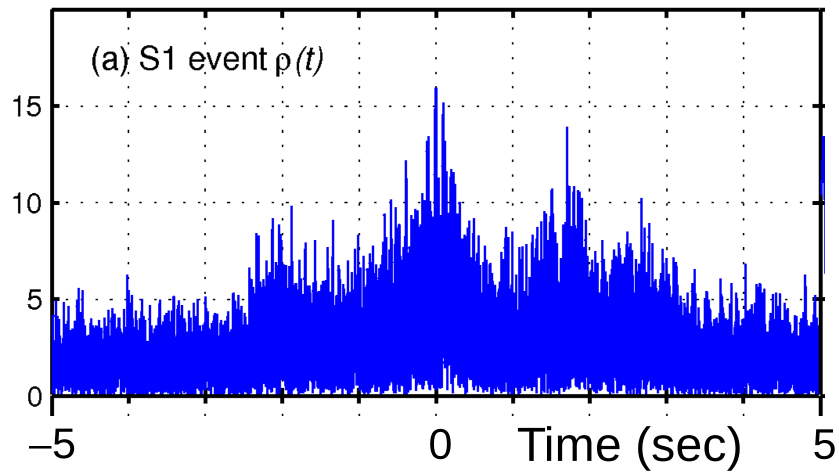
- LIGO.ORG

Extra Slides

Dealing with Non-Stationary Noise



Inspiral
filter output:



Shawhan and Ochsner,
CQG 21, S1757 (2004)

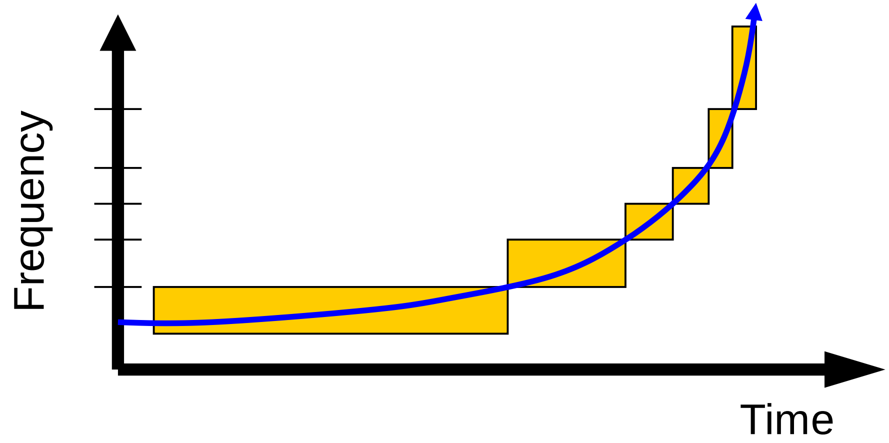
Waveform Consistency Tests

Chi-squared test

Allen, PRD 71, 062001 (2005)

Divide template into p parts, calculate

$$\chi^2(t) = p\sum_{l=1}^{p} \left\| C_l(t) - C(t)/p \right\|^2$$



Can use χ^2 with ρ to form some sort of "effective SNR", e.g.:

$$\rho_{\text{eff}}^2 = \frac{\rho^2}{\sqrt{\left(\frac{\chi^2}{2p-2}\right)\left(1+\frac{\rho^2}{250}\right)}}$$

$$\rho_{\text{new}} = \begin{cases} \rho, & \chi^2 \leq n_{\text{dof}} \\ \dfrac{\rho}{\left[\left(1+\frac{\chi^2}{n_{dof}}^{4/3}\right)/2\right]^{1/4}}, & \chi^2 > n_{\text{dof}} \end{cases}$$

$$\hat{\rho} = \begin{cases} \dfrac{\rho}{[(1+(\chi_r^2)^3)/2]^{1/6}} & \text{for } \chi_r^2 > 1, \\ \rho & \text{for } \chi_r^2 \leq 1. \end{cases}$$

Empirical – to separate signals from background as cleanly as possible