Time-Series Analysis

Introduction

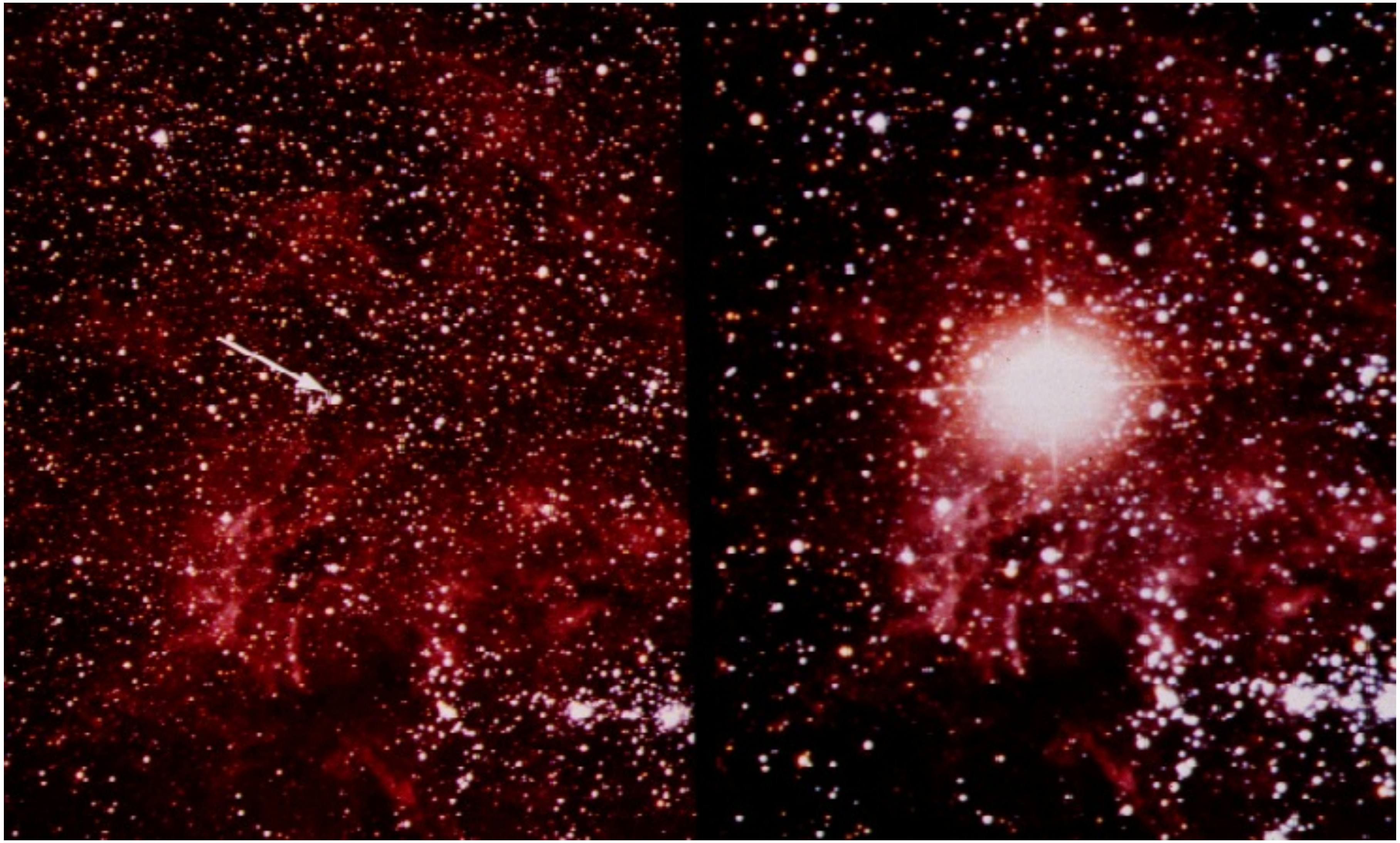
Time-series data in an astronomical context usually consists of:

- unevenly sampled data
- low signal to noise data
- data with heteroscedastic errors

The Gaia satellite will measure billion sources about 70 times and LSST will obtain 800 measurements for about 20 billion sources

Scientific Application of Time-series datasets

- Searches for extra-solar planets
- Searches for astrophysical transients
- Studies of variable stars and supernova explosions
- Distance determination from Cepheids, RR Lyrae
- Tests of GR with radio pulsars
- Search for Gravitational Wave events
- Studies of light-curves of GRBs, supernovae, AGNs



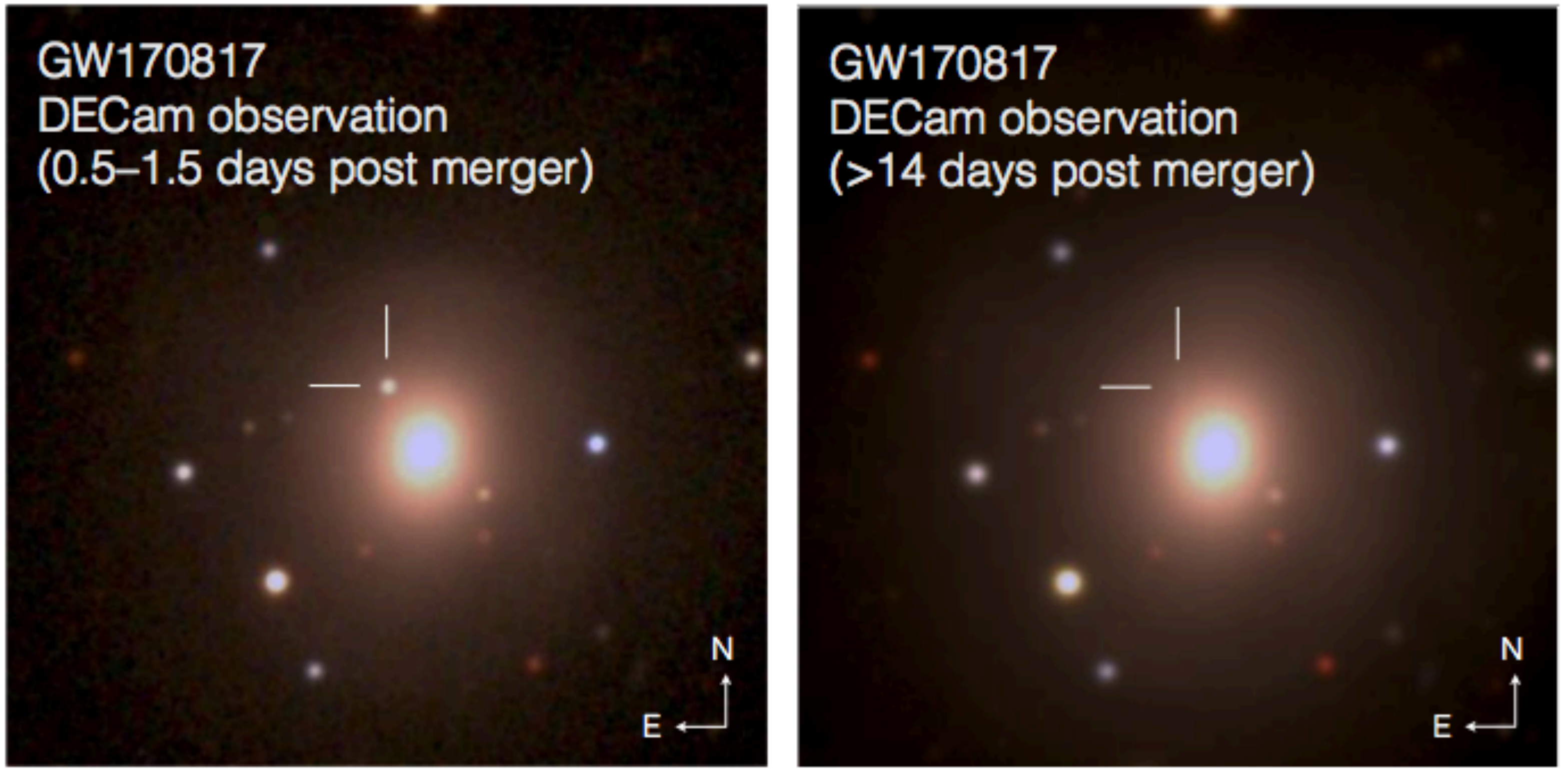
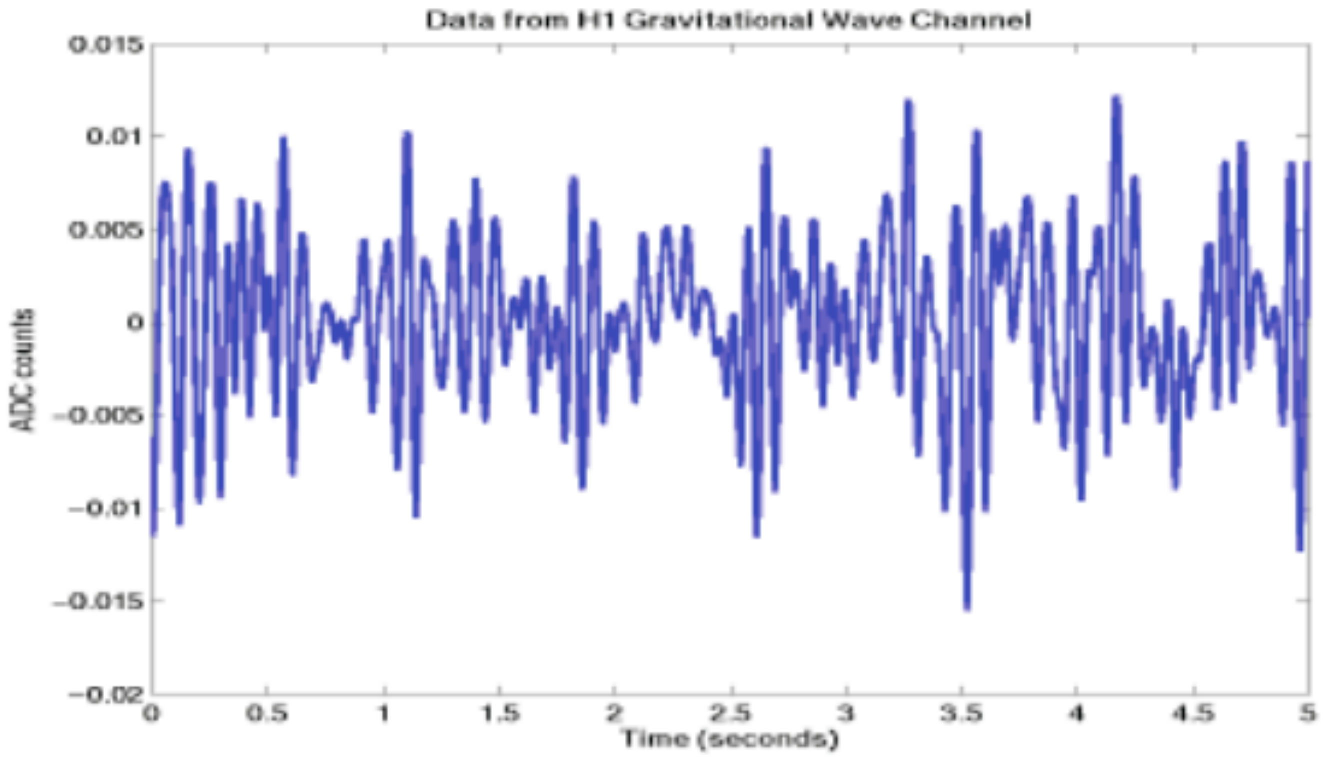
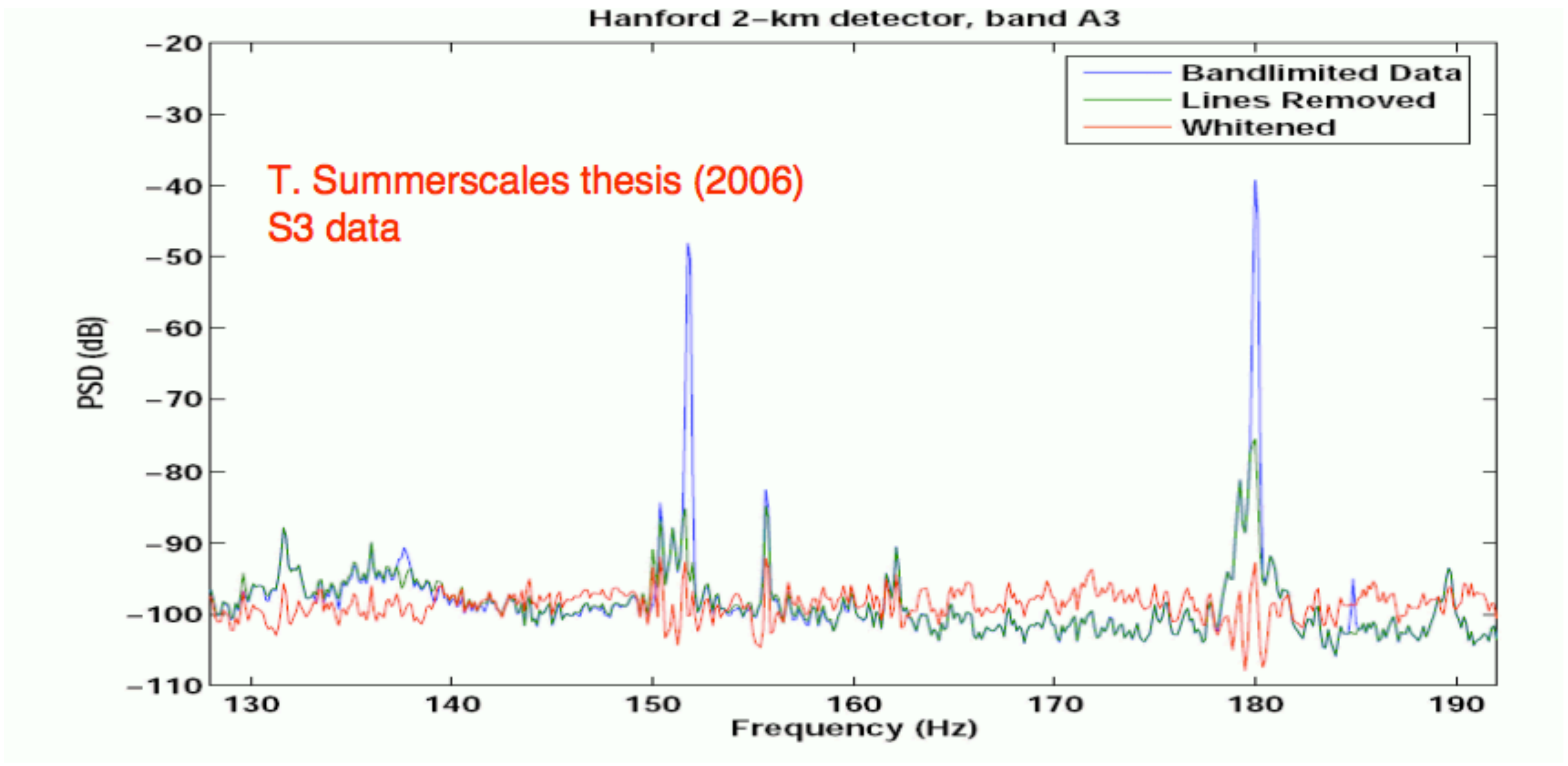


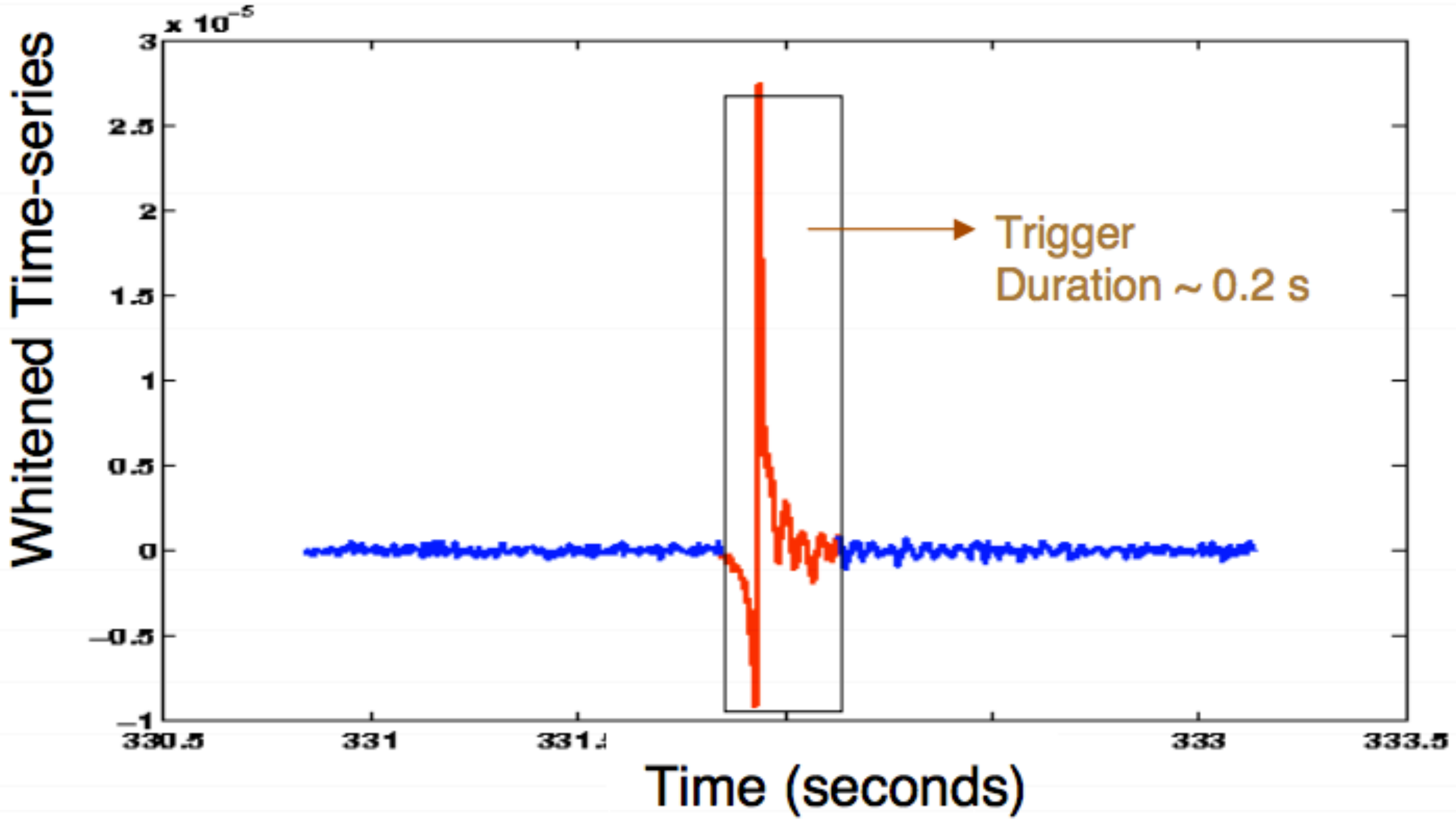
Figure 1. NGC4993 *grz* color composites (1.5'×1.5'). Left: Composite of detection images, including the discovery *z* image taken on 2017 August 18 00:05:23 UT and the g and r images taken 1 day later; the optical counterpart of GW170817 is at RA,Dec = 197.450374, −23.381495. Right: The same area two weeks later.

arXiv:1710.05459

Raw LIGO data




Hanford 2−km detector, band A3
Bandlimited Data
Lines Removed
Whitened
T. Summerscales thesis (2006)
S3 data
PSD (dB)
Frequency (Hz)
−20
−30
−40
−50
−60
−70
−80
−90
−100
−110
130
140
150
160
170
180
190



Looking for transients in LIGO data

Introduction

Time-series datasets contains pairs of random variables belonging to scalar datasets (t_1,y_1), (t_2,y_2)……..(t_n,y_n)

Many analysis methods related to parameter estimation and model comparison (if we replace t by x)

However, some differences in certain types of time-series analysis where directionality is important. In some models values of y_{i+1} depends on the preceding value y_i

Tasks of Time-series Analysis

- Characterize the presumed temporal correlation between different values of y_i including its significance.

- Forecast (predict) future values of y

(solar activity forecasting, prediction of time and place of asteroid impact)

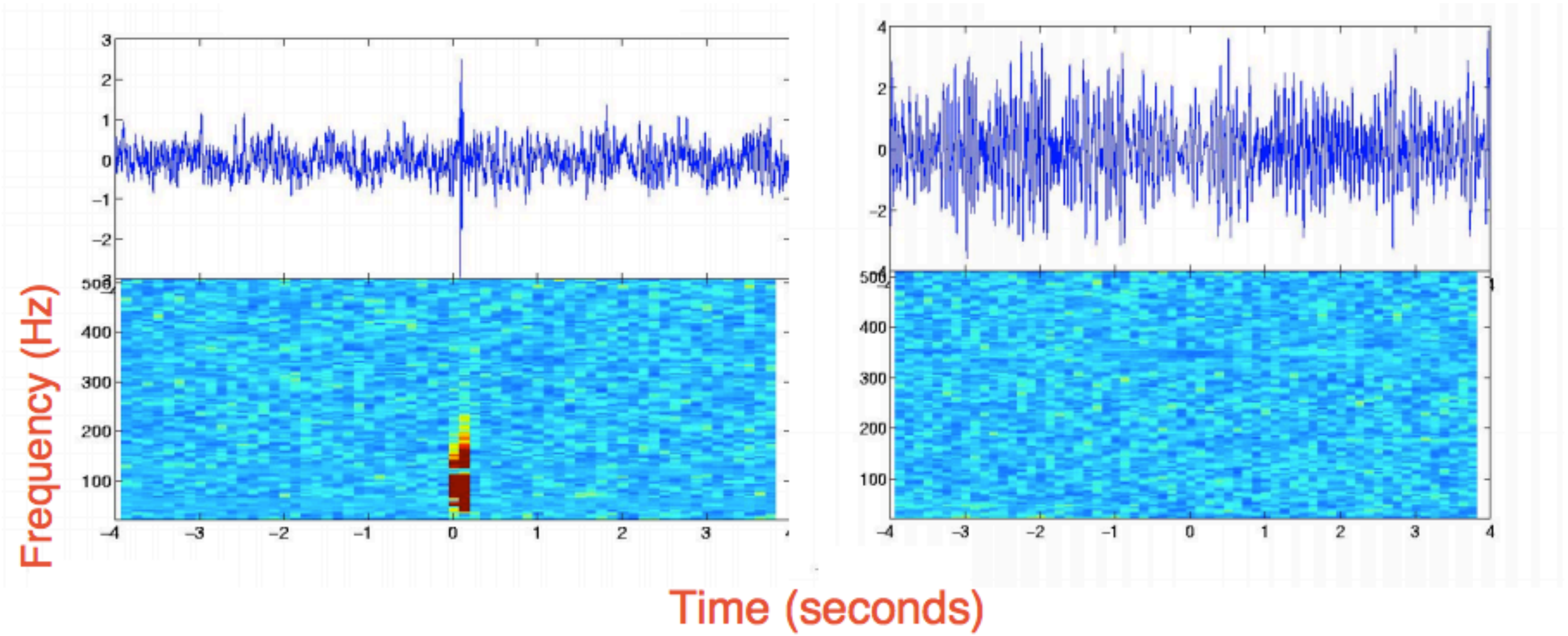
- Characterization of underlying physical processes of the model

(analysis of light curves can distinguish between pulsating and eclipsing stars)

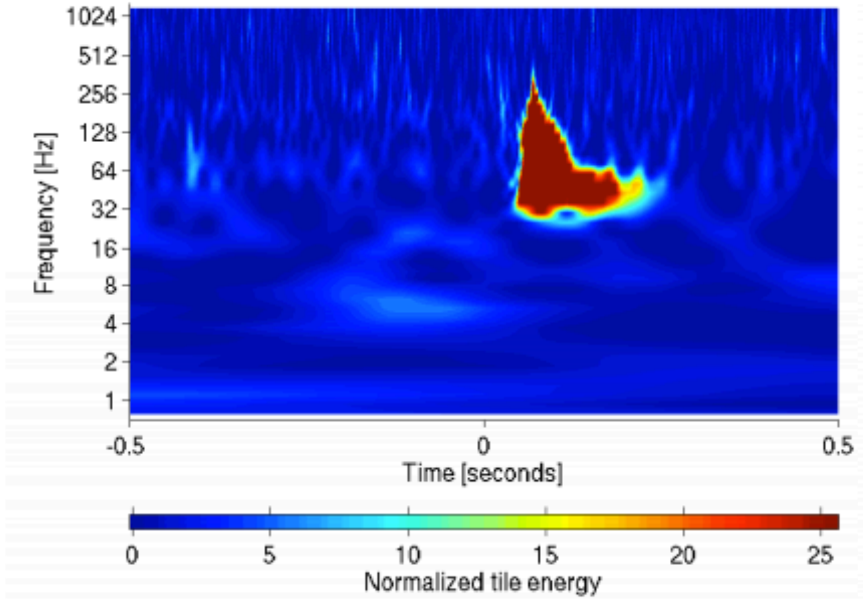
Modelling Toolkit for time-series analysis

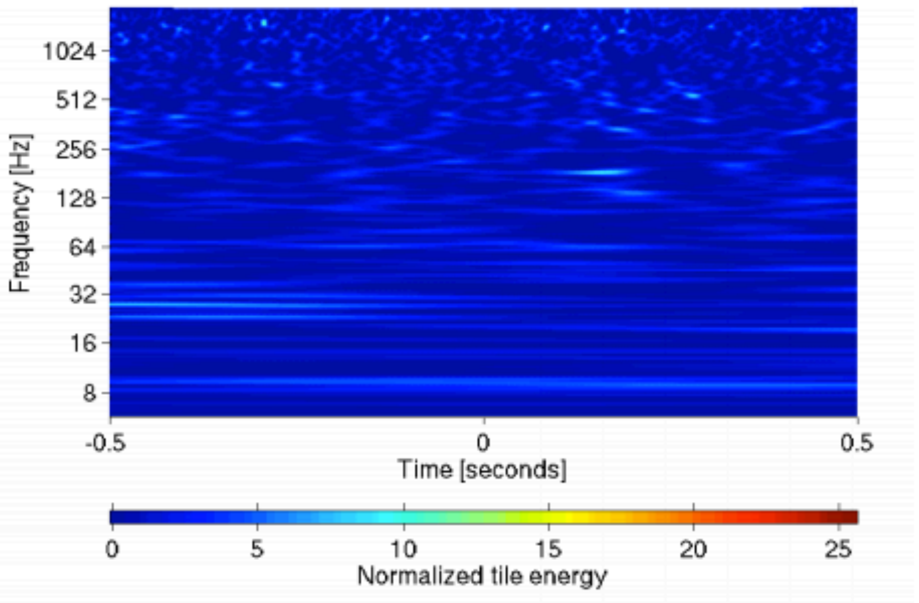
- Time-Domain Method
- Frequency Domain Methods
- Hybrid (Time-Frequency methods) x

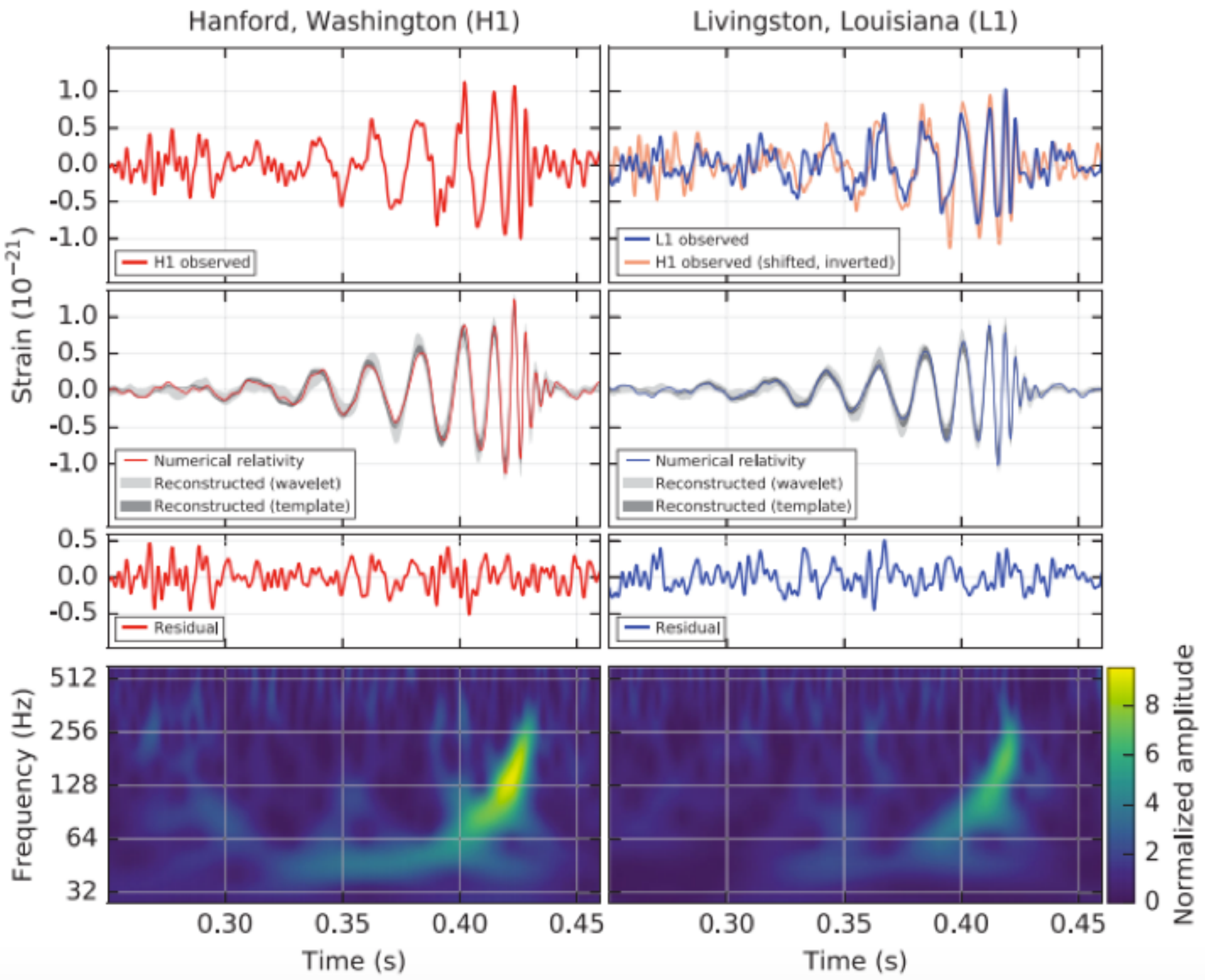
Example of Spectrograms



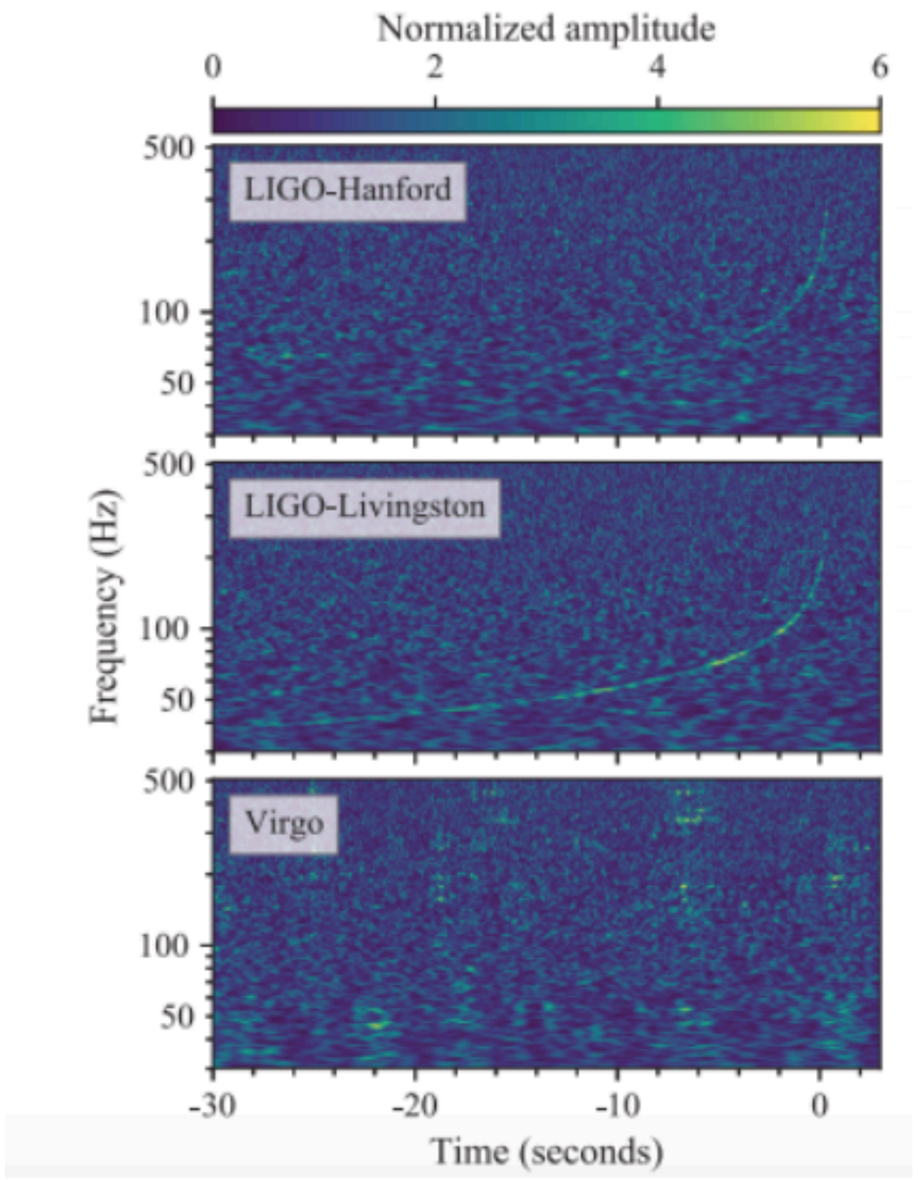
Noise Transient in LIGO data







First LIGO Binary Black Hole signal

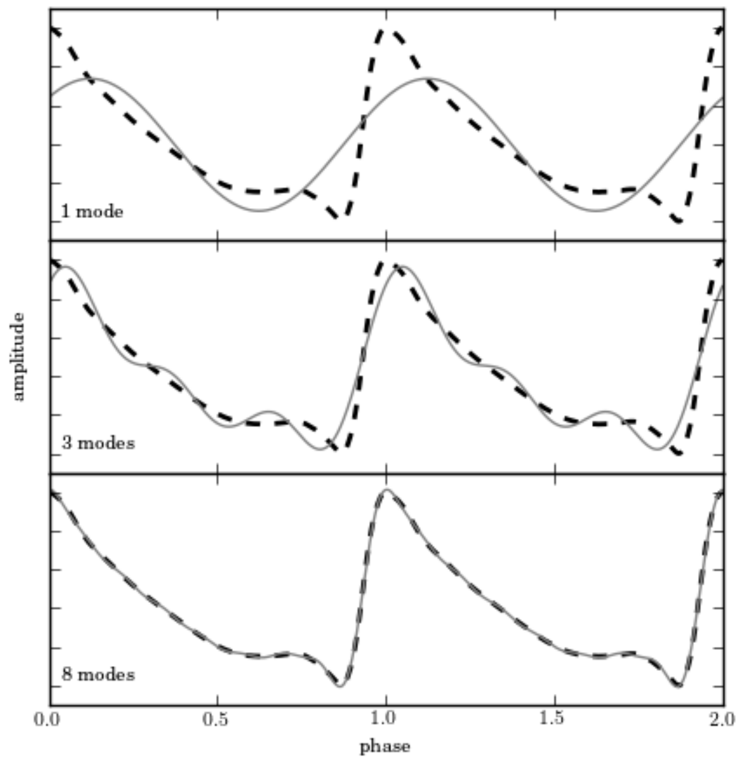


First Binary Neutron Star Gravitational Wave Signal seen by LIGO
GW170817

Application of Fourier Analysis

- Convolution and Deconvolution
- Filtering
- Correlation and Autocorrelation
- Power Spectrum Estimation.

Fourier Series as a Basis



An example of a truncated Fourier representation of an RR Lyrae light curve. The thick dashed line shows the true curve; the gray lines show the approximation based on 1, 3, and 8 Fourier modes (sinusoids).

Definition of Fourier Transform

- Fourier Transform of function h(t) is defined as (scipy convention) :

$$H(f) = \int_{-\infty}^{+\infty} h(t)\exp(-i2\pi ft)dt$$

with inverse transformation defined as :

$$h(t) = \int_{-\infty}^{+\infty} H(f)\exp(i2\pi ft)df$$

Units of H(f) are units of h(t) x inverse Hz.
Some times angular frequency = 2π f is used

- Fourier transform is a complex quantity , except when h(t) is an even function of t

- Fourier transform of a pdf of a zero-mean Gaussian $\mathcal{N}(0, \sigma)$ in the time-domain is a Gaussian given in the frequency domain by

$$H(f) = \exp(-2\pi^2\sigma^2 f^2)$$

- Fourier transform of h (t+ Δt) is given by:

$$\int_{-\infty}^{\infty} h(t + \Delta t) \exp(-i2\pi f \Delta t) dt H(f) \exp(i2\pi f \Delta t)$$

- Fourier Transform of a Gaussian $\mathcal{N}(\mu, \sigma)$ is given by :

$$H_{Gauss}(f) = \exp(-2\pi^2\sigma^2 f^2)[\cos(2\pi f\mu) + i\sin(2\pi f\mu)]$$

This is different than the F.T. of Gaussian noise with time-independent variance which is simply a constant and referred to as white noise.

One-sided PSD is defined as : $PSD(f) = |H(f)|^2 + |H(-f)|^2$
PSD gives the amount of power contained in the frequency interval between *f* and *f+df*

When h(t) = sin(2πt/T) , P(f) is a delta function centered on f=1/T

Parseval's Theorem

Total power is the same whether computed in the frequency domain or in the time domain:

$$P_{tot} = \int_0^\infty PSD(f)df = \int_{-\infty}^\infty |h(t)|^2 dt$$

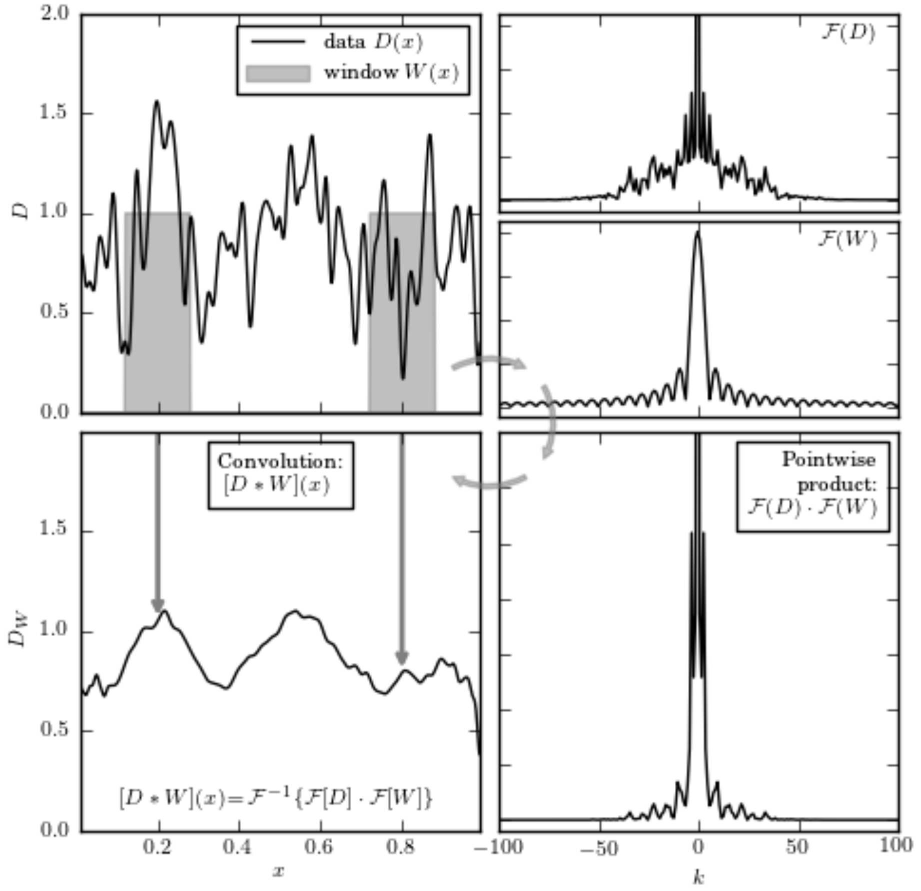
Conservation of Energy

Convolution Theorem

Convolution of two functions a(t) and b(t) is given by:

$$(a \star b)(t) = \int_{-\infty}^{\infty} a(t')b(t-t')dt'$$

F.T. of the above equations are given by:
H(f) = A(f) B(f)



A schematic of how the convolution of two functions works. The top-left panel shows simulated data (black line); this time series is convolved with a top-hat function (gray boxes); see eq. 10.8. The top-right panels show the Fourier transform of the data and the window function. These can be multiplied together (bottom-right panel) and inverse transformed to find the convolution (bottom-left panel), which amounts to integrating the data over copies of the window at all locations. The result in the bottom-left panel can be viewed as the signal shown in the top-left panel smoothed with the window (top-hat) function.

Discrete Fourier Transform

In real life data are always discretely sampled.
Assume h(t) is sampled at N equal time-intervals $h_j = h(t_j)$ with $t_j = t_0 + j\Delta t$, with $j = 0,\ldots(N-1)$ and $T = N\Delta t$

$$H_k = \sum_{j=0}^{N-1} h_j \exp[-i2\pi jk/N]$$ where k= 0…..(N-1)

Inverse Discrete Fourier Transform is defined by:

$$h_j = \frac{1}{N}\sum_{k=0}^{N-1} H_k \exp[i2\pi jk/N]$$ where j= 0,……(N-1).

Nyquist Sampling Theorem

Define h(t) to be *band-limited* if H(f) = 0 for |f| >|f_c| where f_c is the band limit or the Nyquist critical frequency. Define t_c=1/(2f_c)

According to Nyquist sampling theorem, we can *exactly reconstruct* h(t) from evenly sampled data when Δt <= t_c as follows:

$$h(t) = \frac{\Delta t}{t_c} \sum_{k=-\infty}^{\infty} h_k \frac{\sin[2\pi f_c(t - k\Delta t)]}{2\pi f_c(t - k\Delta t)}$$

If h(t) = sin(2π t/P), Period = P and is band-limited with f_c=1/P
If it is sampled at Δt <= P/2 it can be fully reconstructed at any t

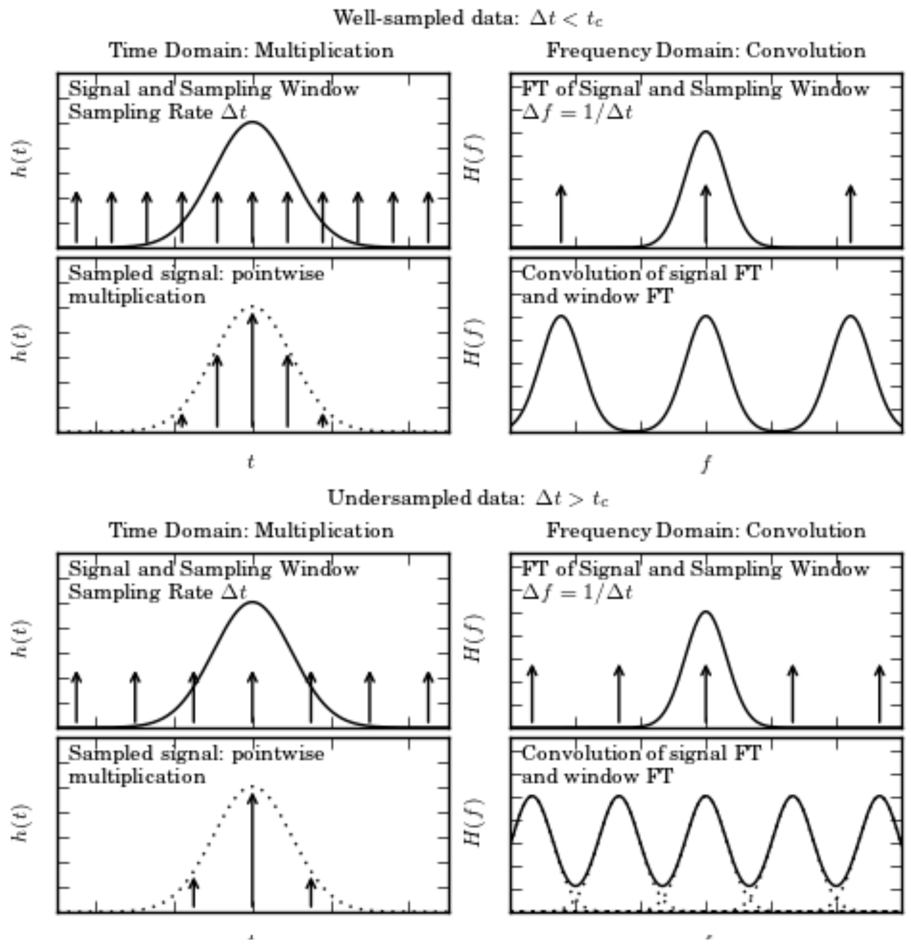
Aliasing

- If the signal is not band-limited or when sampling rate is not sufficient ($\Delta t > t_c$) an effect called aliasing prevents us from exactly reconstructing h(t).

- In such cases all P.S.D. from frequencies $|f| > f_c$ is aliased (falsely transferred) into the $-f_c < f < f_c$ range.

- Aliasing can be thought of as inability to resolve details in a time-series

Aliasing can be recognized if the Fourier transform is non-zero at $|f| = 1/(2\ \Delta t)$


Well-sampled data: $\Delta t < t_c$
Time Domain: Multiplication
Frequency Domain: Convolution
Signal and Sampling Window
Sampling Rate Δt
FT of Signal and Sampling Window
$\Delta f = 1/\Delta t$
Sampled signal: pointwise multiplication
Convolution of signal FT and window FT
$h(t)$
$H(f)$
t
f
Undersampled data: $\Delta t > t_c$
Time Domain: Multiplication
Frequency Domain: Convolution
Signal and Sampling Window
Sampling Rate Δt
FT of Signal and Sampling Window
$\Delta f = 1/\Delta t$
Sampled signal: pointwise multiplication
Convolution of signal FT and window FT

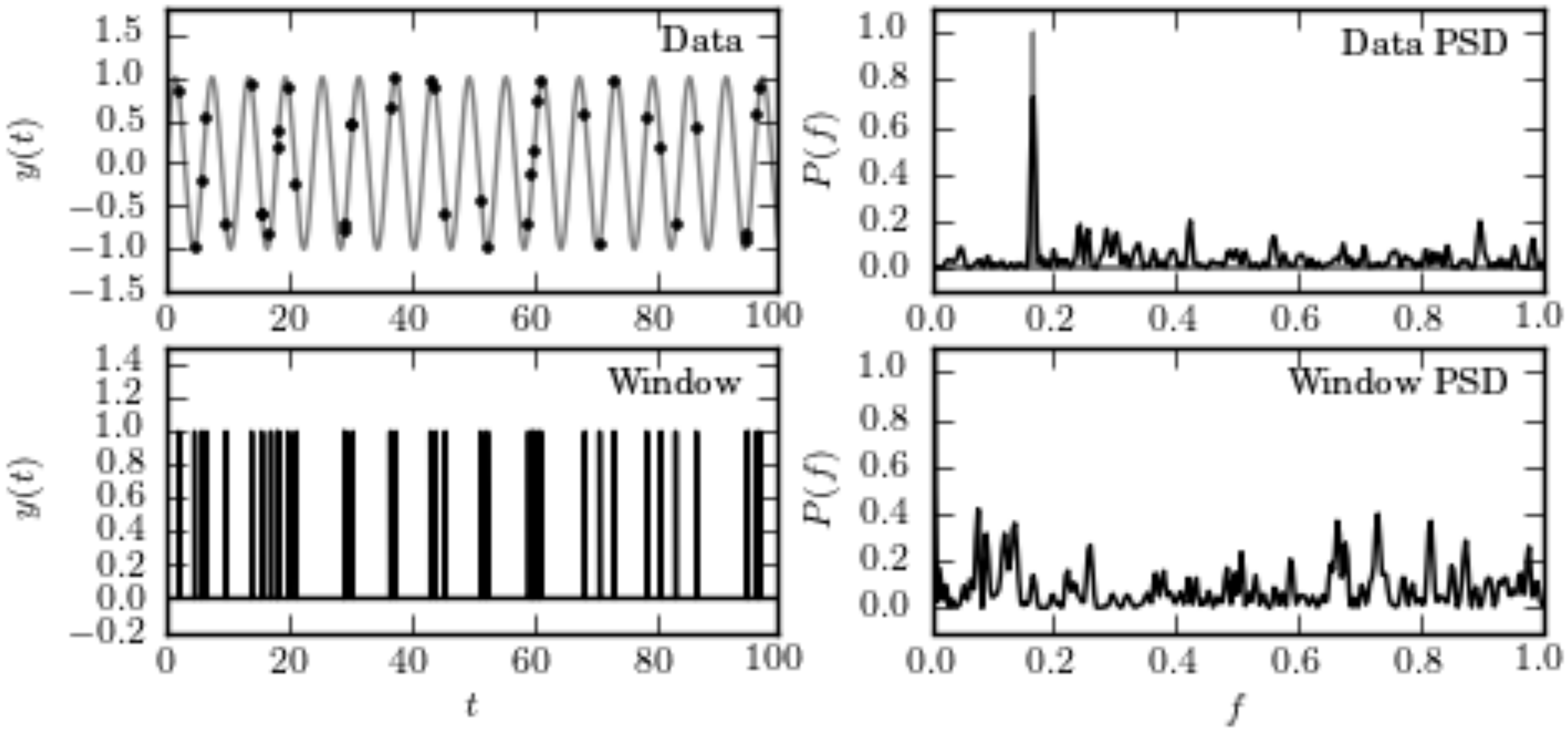
A visualization of aliasing in the Fourier transform. In each set of four panels, the top-left panel shows a signal and a regular sampling function, the top-right panel shows the Fourier transform of the signal and sampling function, the bottom-left panel shows the sampled data, and the bottom-right panel shows the convolution of the Fourier-space representations (cf. figure 10.2). In the top four panels, the data is well sampled, and there is little to no aliasing. In the bottom panels, the data is not well sampled (the spacing between two data points is larger) which leads to aliasing, as seen in the overlap of the convolved Fourier transforms (figure adapted from Greg05).

Window Functions

Sampling window function in time-domain can be expressed as sum of delta functions placed at sampled observation times.
F.T. of set of delta functions with spacing Δt is another set of delta functions with Spacing 1/Δt

When data are not uniformly sampled, impact of sampling can be understood in a similar fashion. Sampling window is a sum of delta functions (not regularly spaced).

Observed PSD is a convolution of the true underlying PSD and spectral window Function.

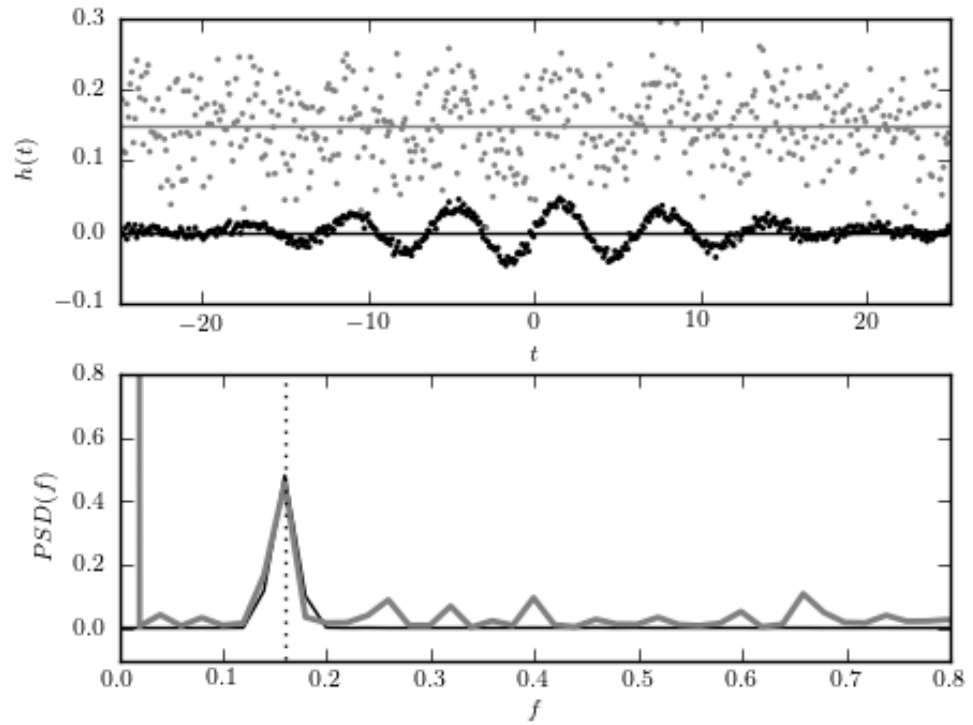


An illustration of the impact of a sampling window function of resulting PSD. The top-left panel shows a simulated data set with 40 points drawn from the function y(tlP) = sin(t) (i.e., f = 1/(2pi) ~ 0.16). The sampling is random, and illustrated by the vertical lines in the bottom-left panel. The PSD of sampling times, or spectral window, is shown in the bottom-right panel. The PSD computed for the data set from the top-left panel is shown in the top-right panel; it is equal to a convolution of the single peak (shaded in gray) with the window PSD shown in the bottom-right panel (e.g., the peak at f ~ 0.42 in the top-right panel can be traced to a peak at f ~ 0.26 in the bottom-right panel).

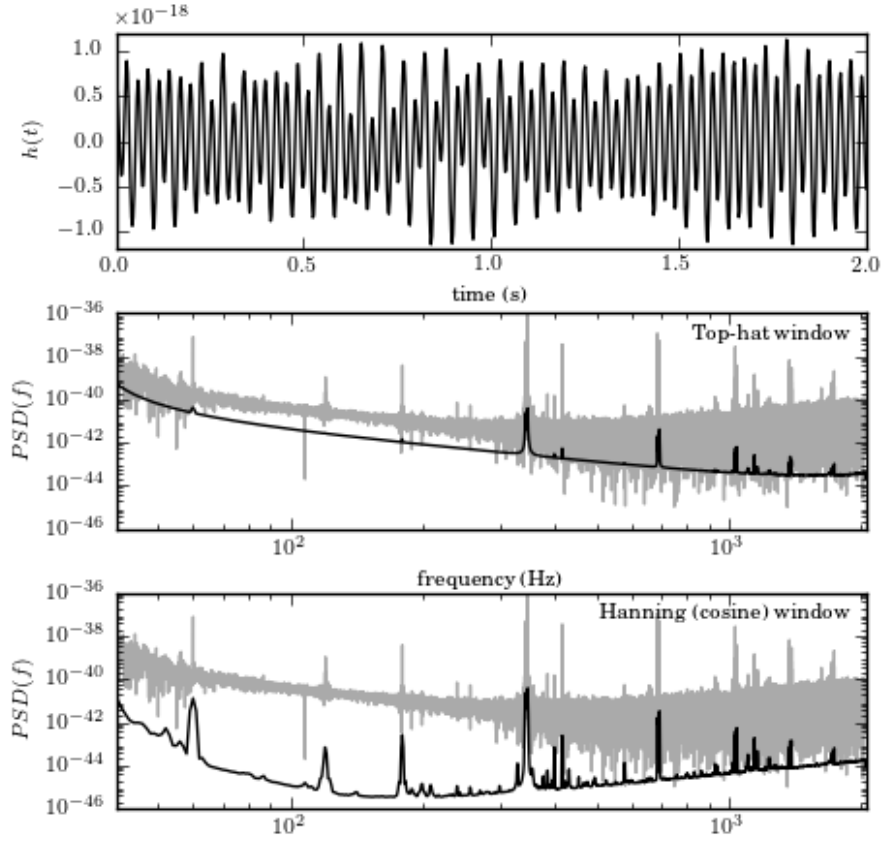
Fast Fourier Transform

FFT is an an algorithm for computing discrete Fourier Transform in O(N logN) time instead of O(N^2)

```
import numpy as np
from scipy import fftpack
x=np.random.random(size=1000)
X_fft=fftpack(x) #Fourier transform
x2= fftpack.ifft(x_fft) #inverse :x2=x to numerical precision
```

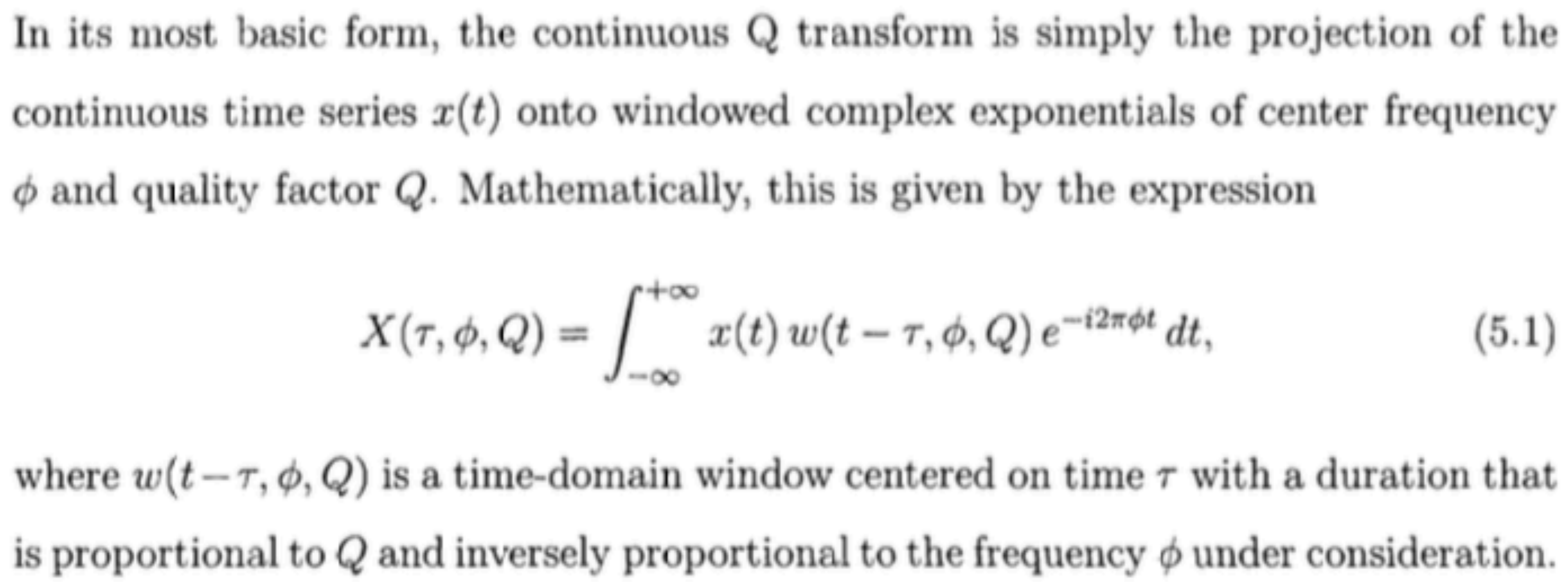


The discrete Fourier transform (bottom panel) for two noisy data sets shown in the top panel. For 512 evenly sampled times t (dt = 0.977), points are drawn from h(t) = a + sin(t)G(t), where G(t) is a Gaussian N(mu = 0,sigma = 10). Gaussian noise with sigma = 0.05 (top data set) and 0.005 (bottom data set) is added to signal h(t). The value of the offset a is 0.15 and 0, respectively. The discrete Fourier transform is computed as described in Section 10.2.3. For both noise realizations, the correct frequency f = (2pi)-1 ~ 0.159 is easily discernible in the bottom panel. Note that the height of peaks is the same for both noise realizations. The large value of abs(H(f = 0)) for data with larger noise is due to the vertical offset.



LIGO data and its noise power spectrum. The upper panel shows a 2-second-long stretch of data (~8000 points; essentially noise without signal) from LIGO Hanford. The middle and bottom panels show the power spectral density computed for 2048 seconds of data, sampled at 4096 Hz (~8 million data values). The gray line shows the PSD computed using a naive FFT approach; the dark line uses Welch's method of overlapping windows to smooth noise; the middle panel uses a 1-second-wide top-hat window and the bottom panel the so-called Hanning (cosine) window with the same width.

Q-Transform



In its most basic form, the continuous Q transform is simply the projection of the continuous time series $x(t)$ onto windowed complex exponentials of center frequency ϕ and quality factor Q. Mathematically, this is given by the expression

$$X(\tau,\phi,Q) = \int_{-\infty}^{+\infty} x(t)\, w(t-\tau,\phi,Q)\, e^{-i2\pi\phi t}\, dt, \tag{5.1}$$

where $w(t-\tau,\phi,Q)$ is a time-domain window centered on time τ with a duration that is proportional to Q and inversely proportional to the frequency ϕ under consideration.

Shourov Chatterji Ph.D thesis MIT (2005)

Wavelets

- Trigonometric basis functions used for the Fourier transform have an infinite extent and for this reason the Fourier transform may not be the best method to analyze non-periodic time-series data (for eg. a localized event)

Choose basis functions that are localized themselves (e.g. wavelets)

Wavelets are localized in *both* frequency and time domains. Individual wavelets are specified by a set of wavelet filter coefficients. Given a wavelet, a complete orthonormal set of basis functions can be constructed by scalings and transformations.

Some examples of wavelets are:

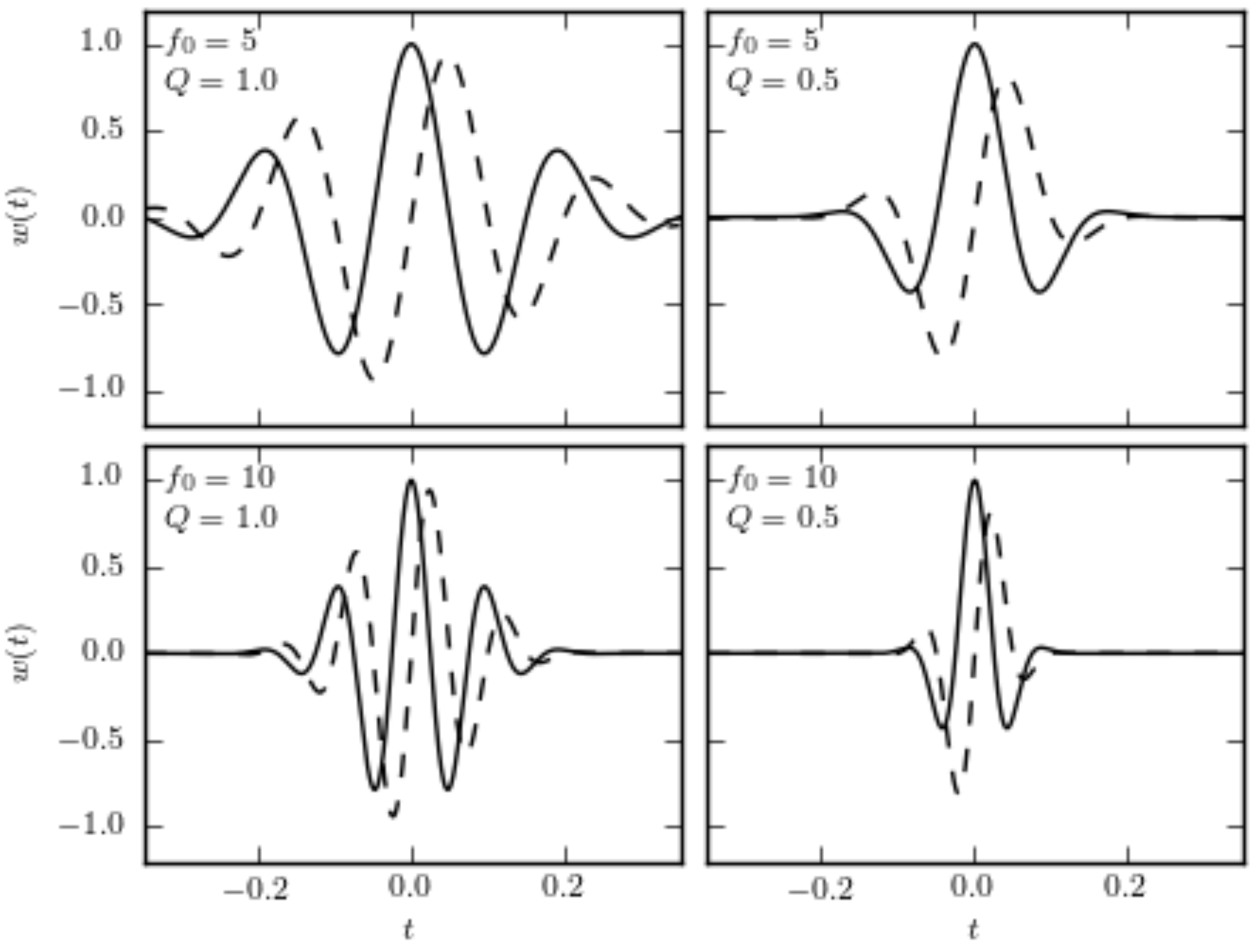
- Daubechies wavelets
- Mexican hat wavelets
- Haar wavelets

Example of Wavelet

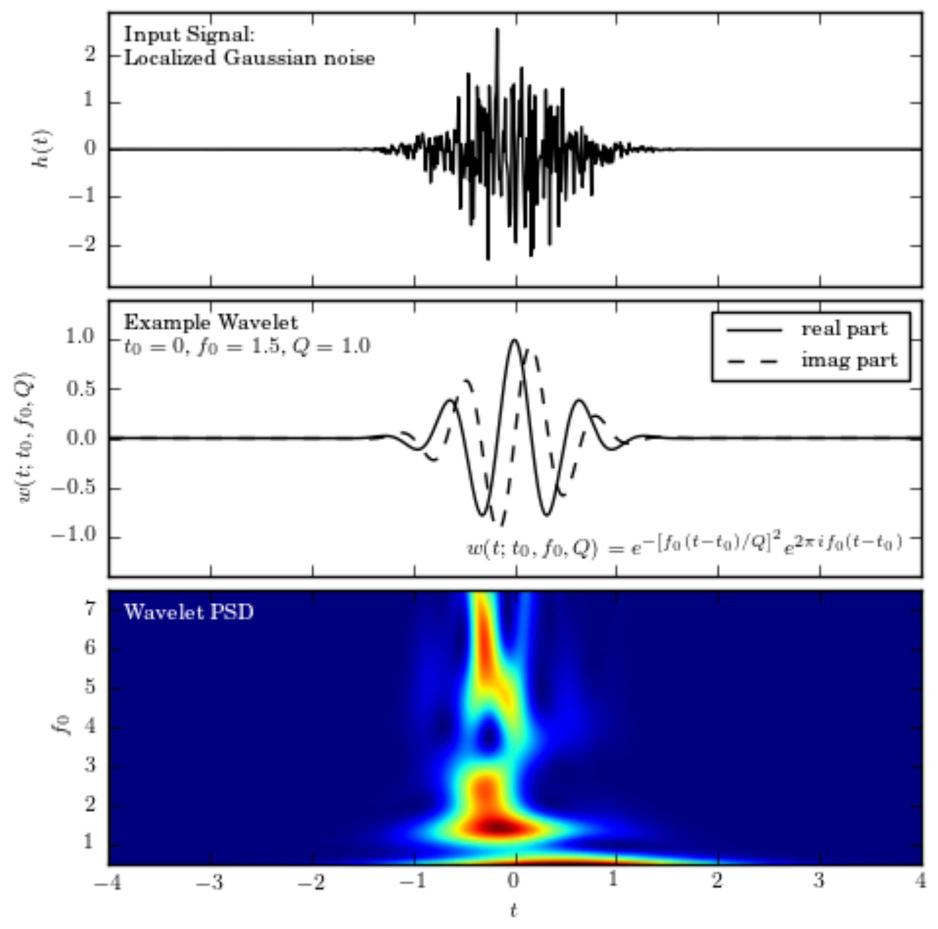
$$w(t|t_0, f_0, Q) = A\exp[i2\pi f_0(t - t_0)]\exp[-f_0^2(t - t_0)^2/Q^2]$$

where t_0 is the central time and f_0 is the central frequency and Q is a model parameter which controls the width of the frequency window.

(This is closely related to Morlet wavelet)



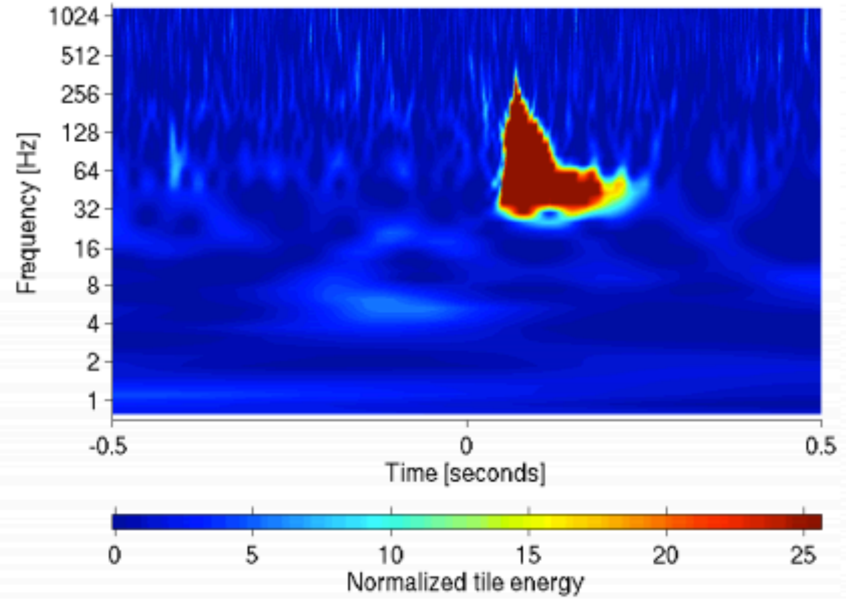
Wavelets for several values of wavelet parameters Q and f0. Solid lines show the real part and dashed lines show the imaginary part (see eq. 10.16).

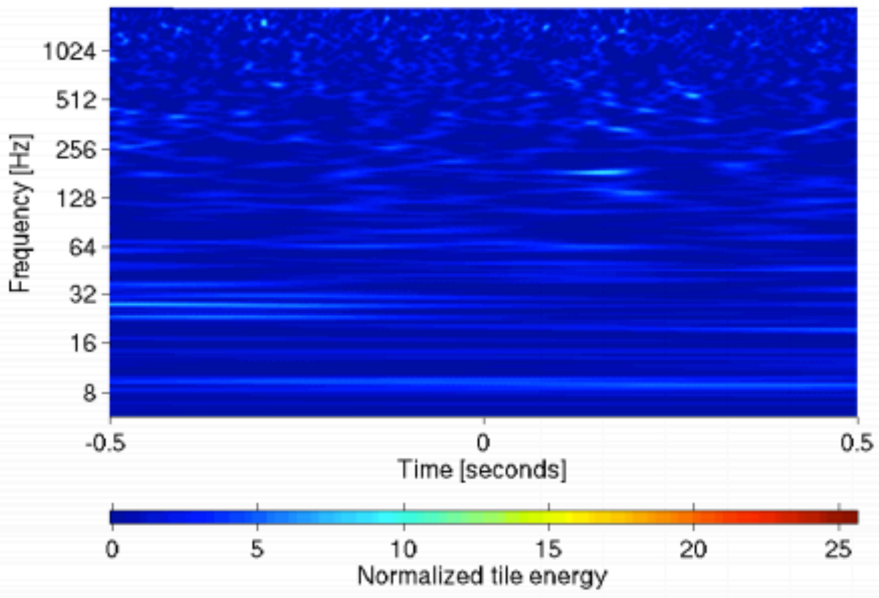


AstroML figure 10.7

Localized frequency analysis using the wavelet transform. The upper panel shows the input signal, which consists of localized Gaussian noise. The middle panel shows an example wavelet. The lower panel shows the power spectral density as a function of the frequency f0 and the time t0, for Q = 1.0.

Example of Q-Transform in LIGO data





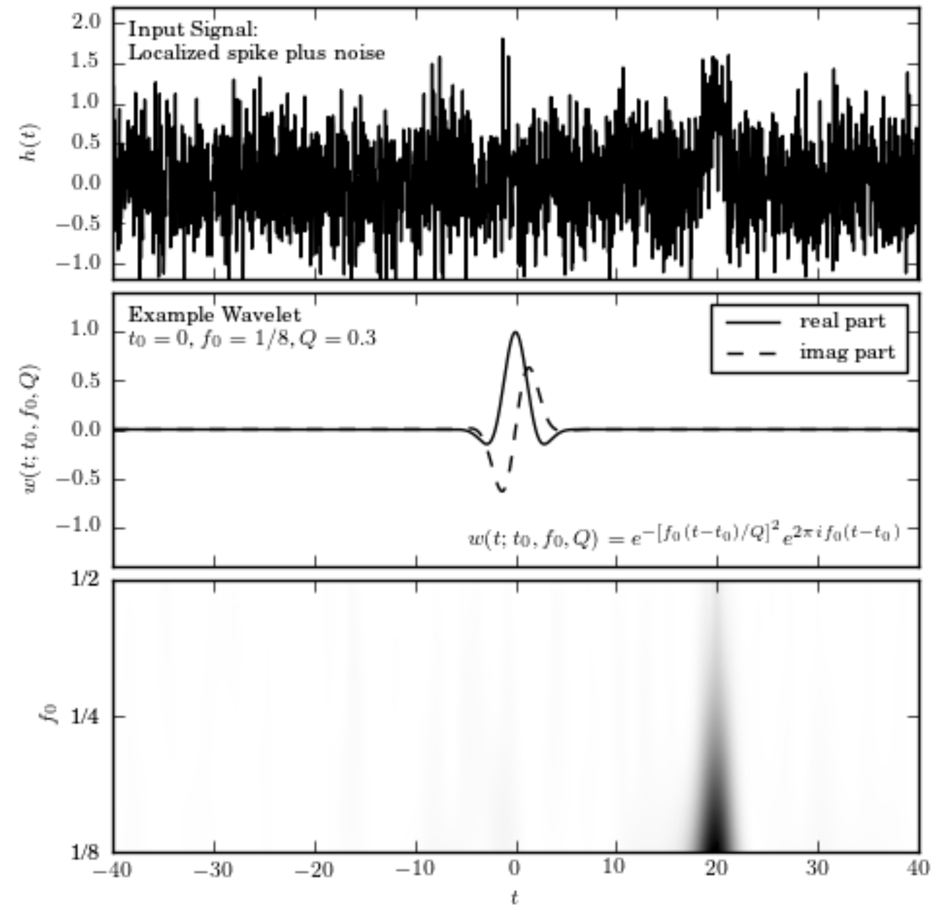
Wavelet Transform

Wavelet transform applied to the data h(t) is given by :

$$H_w(t_0; f_0, Q) = \int_{-\infty}^{+\infty} h(t) w(t|t_0, f_0, Q)$$

Wavelet PSD is given by : $PSD_w(f_0, t_0; Q) = |H_w(t_0; f_0, Q)|^2$

Wavelet PSD allows detection of frequency information which is localized in time.
LIGO uses functions tuned to the expected form of the signal (i.e. matched filters)



Localized frequency analysis using the wavelet transform. The upper panel shows the input signal, which consists of a Gaussian spike in the presence of white (Gaussian) noise (see figure 10.10). The middle panel shows an example wavelet. The lower panel shows the power spectral density as a function of the frequency f0 and the time t0, for Q = 0.3.

Wavelets in Python

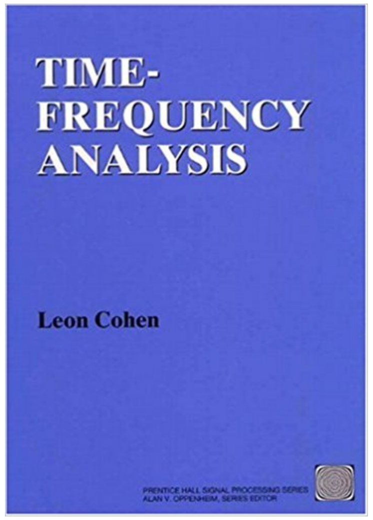
```
import numpy as np
from  astroML.fourier import wavelet_PSD
t=np.linspace(0,1,1000) # times of signal
x=np.random.normal(size=1000) # white noise
f0 = np.linspace(0.01,1,100) # candidate frequencies

WPSD = wavelet_PSD(t,x,f0, Q=1) # 100 x 1000 PSD
```

Beyond Wavelets

- Matching Pursuit, which utilizes large a large redundant set of non-orthogonal functions. Data themselves are used to derive appropriate basis functions.
- Hilbert-Huang transform
- Wigner-Ville Transform

A plethora of such methods are discussed in


TIME-
FREQUENCY
ANALYSIS

Leon Cohen

PRENTICE HALL SIGNAL PROCESSING SERIES
ALAN V. OPPENHEIM, SERIES EDITOR