# Application of NHA to detector characterization

19 Aug. 2014

K. Ueno (Osaka Univ.)

# Non-Harmonic Analysis (NHA)

- NHA is a method to extract dominant frequency components in given time-series with high resolution, developed by a Toyama Univ. group which is led by Hirobayashi-san.

- The method itself is patent-protected and so not publicly available (even to the other KAGRA collaborators), but there are some papers which describe the outline of the method, especially,
  - “Noise reduction for periodic signals using high resolution frequency analysis” Yoshizawa et al., EURASIP Journal on Audio, Speech, and Music Processing 5, 2011

# Basic Algorithm

1) FFT the given time series x(n) and find the frequency which gives the largest amplitude.

2) You somehow minimize the following cost function about A,f, and ϕ,

$$F(A, f, \varphi) = \frac{1}{N}\sum_{n=0}^{N-1}\left\{x(n) - A\cos\left(2\pi\frac{f}{f_s}n + \varphi\right)\right\}^2$$

   starting from A and f estimated at 1). This is just a least square fit with a sinusoidal function.

3) Once the best-fit values of A,f, and ϕ are found, the waveform of converged spectrum is subtracted from x(n).

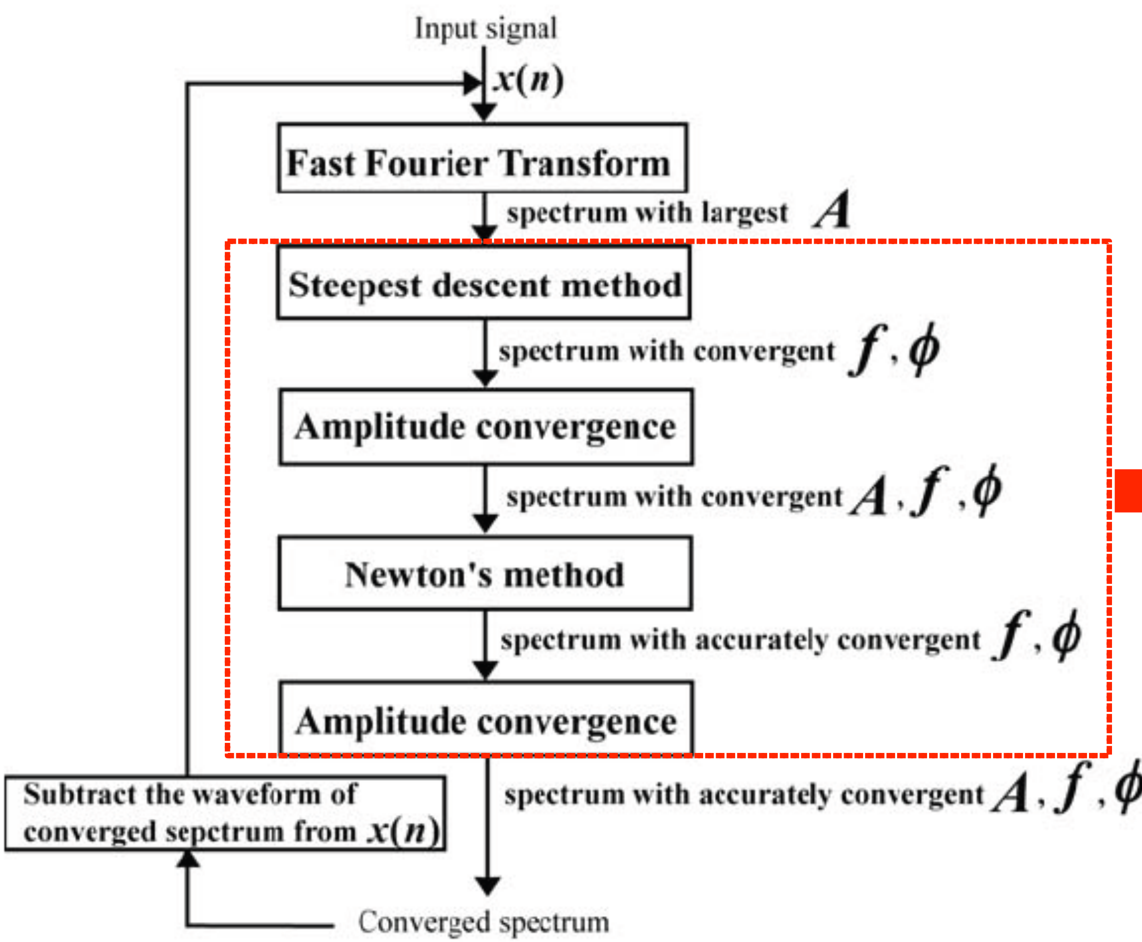
4) Repeat the procedure 1 〜 3 as many times as one would like.

# Current status

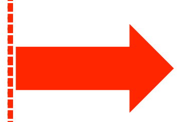
- According to Hirobayashi-san, you need some specific environments such as MATLAB or GPGPU to execute their NHA code.
- In order to do data analysis ourselves efficiently, it's better to develop our own simple version of the code instead of using their real one.
- So I tried to implement this method from the scratch. But when I followed the paper faithfully, the solutions were found not to be stable and soon diverge.
- Instead of following the whole procedure written in the paper, I took another way to meet the same purpose. I'll refer to the new way (next slide) as "Iterative Least Square (ILS)" instead of NHA to avoid confusion.

# Difference from NHA's paper

NHA paper




Iterative Least Square (ILS)



At one time with
3D Newton's method

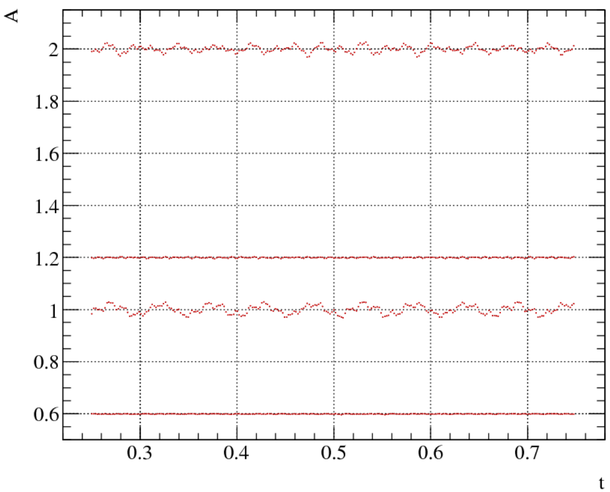
# Test Example

- To test the performance of the ILS code, I artificially generated a signal time-series with some input parameters and tried to reconstruct them with ILS.
- The time-series is composed of four different signals, each of which has a constant amplitude and frequency.
- The sampling rate is 512 Hz, and the duration is set to 1 s.

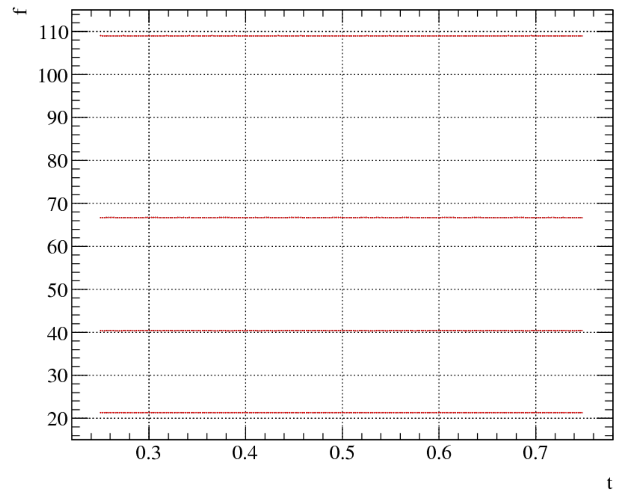
| Amplitude | Frequency [Hz] | Initial phase [rad] |
|---|---|---|
| 2.0 | 66.7 | -0.15 |
| 1.2 | 109.0 | 0.46 |
| 1.0 | 40.4 | 2.40 |
| 0.6 | 21.3 | 0.10 |

# Result (absolute)

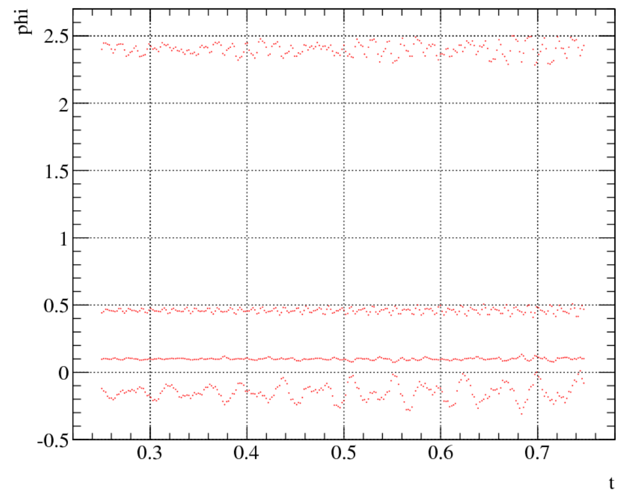
Amplitude




Frequency




Initial phase

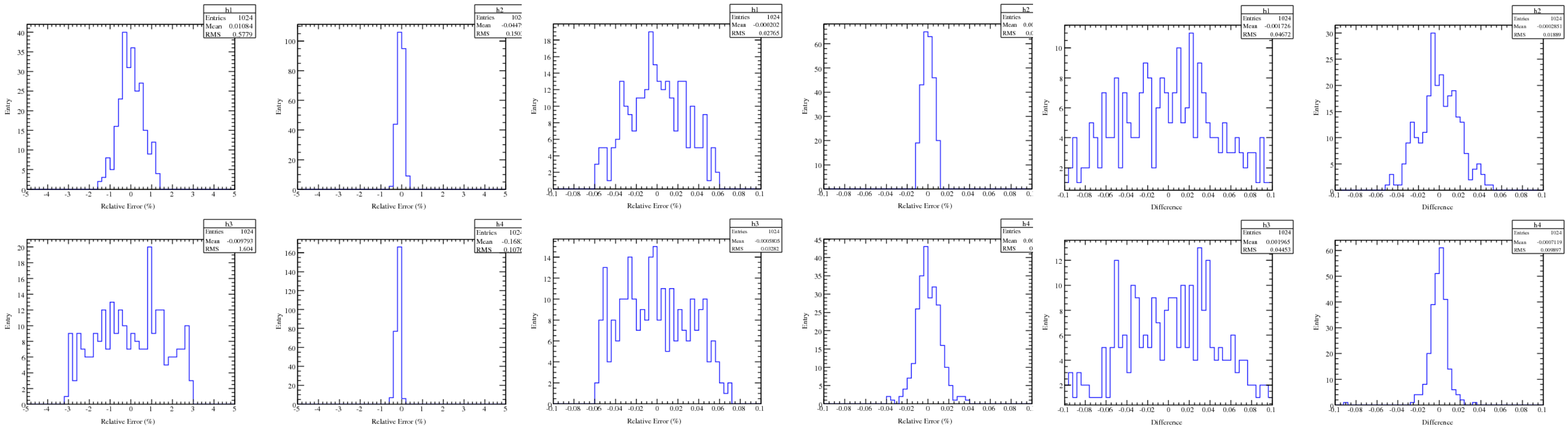



While frequencies were well reconstructed, some of the amplitudes were not, which are caused by signal-signal interference.

What is already clear: when there are multiple signals within the same duration, the minimum of the cost function does not always correspond to the true amplitude value, and the solution shift a little bit.

# Result (resolution)

Amplitude Frequency Initial phase



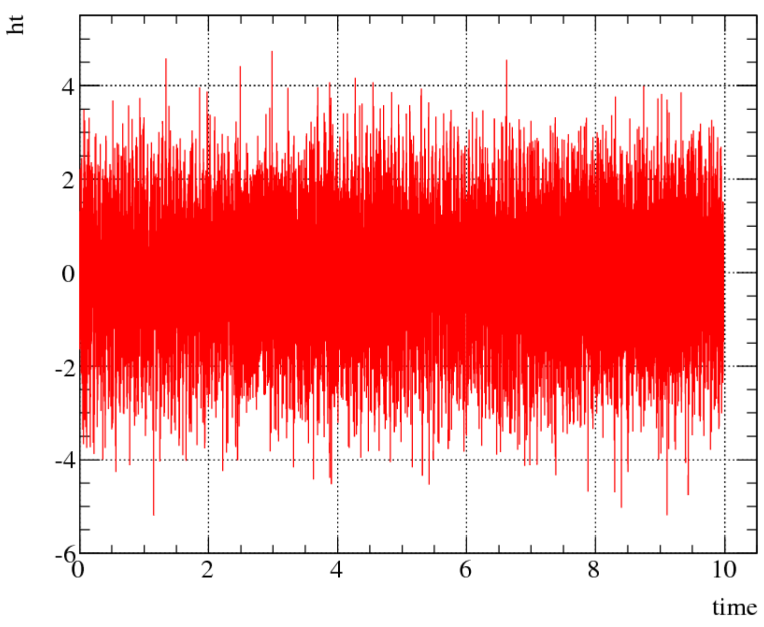

Frequencies were reconstructed within 0.1%, while amplitudes 3%.

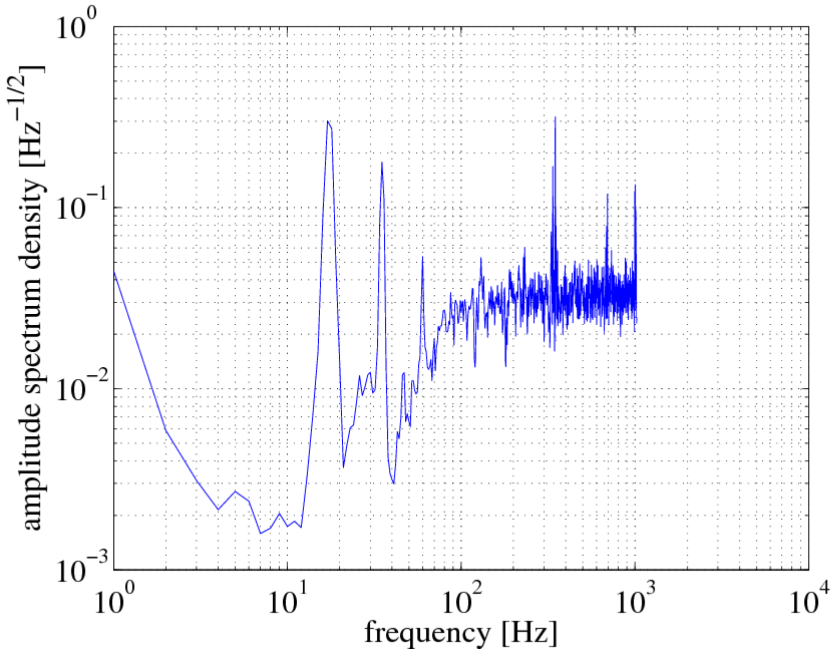
Though the ILS code has plenty of room to improve, let's apply this technology to LIGO real data!

# LIGO S5 (H1) data

Hayama-san gave me a sample time-series from LIGO S5 (H1) data and its noise spectrum.

T=10 sec, fs=2048 Hz

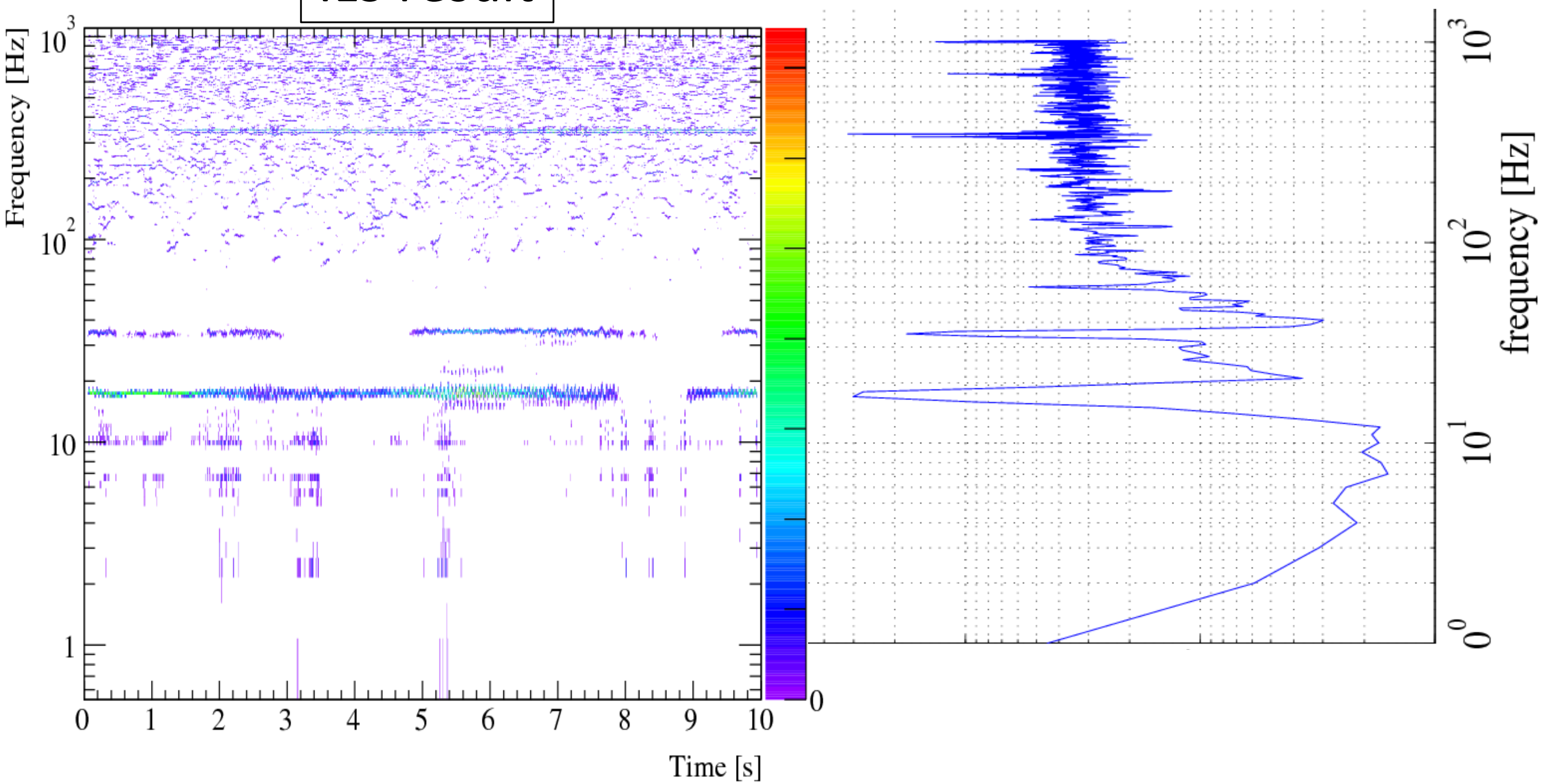



The ordinary noise spectrum tells us the long-time average of the line amplitude and frequency.

# NHA can reveal fine structure of the lines

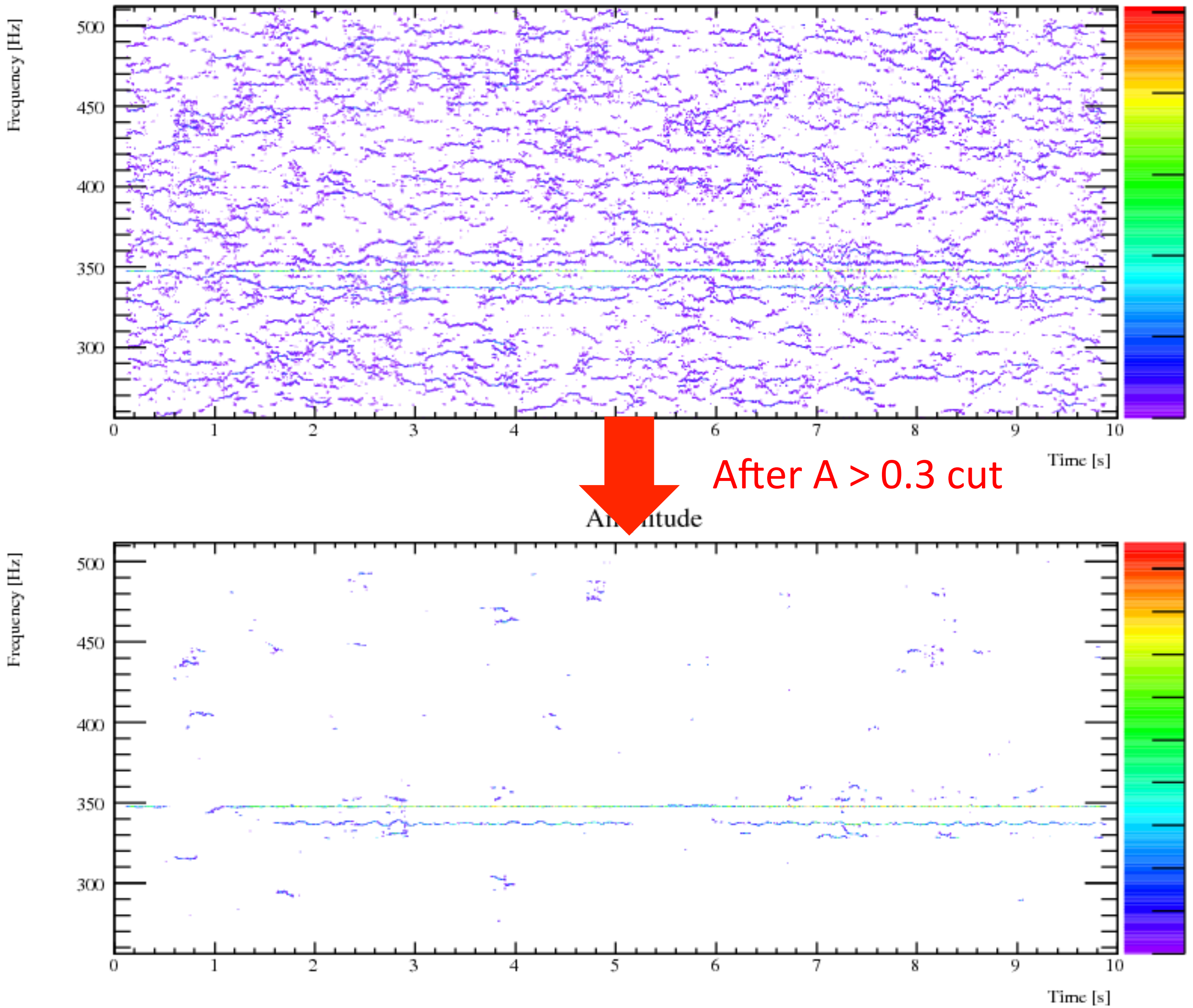
ILS result



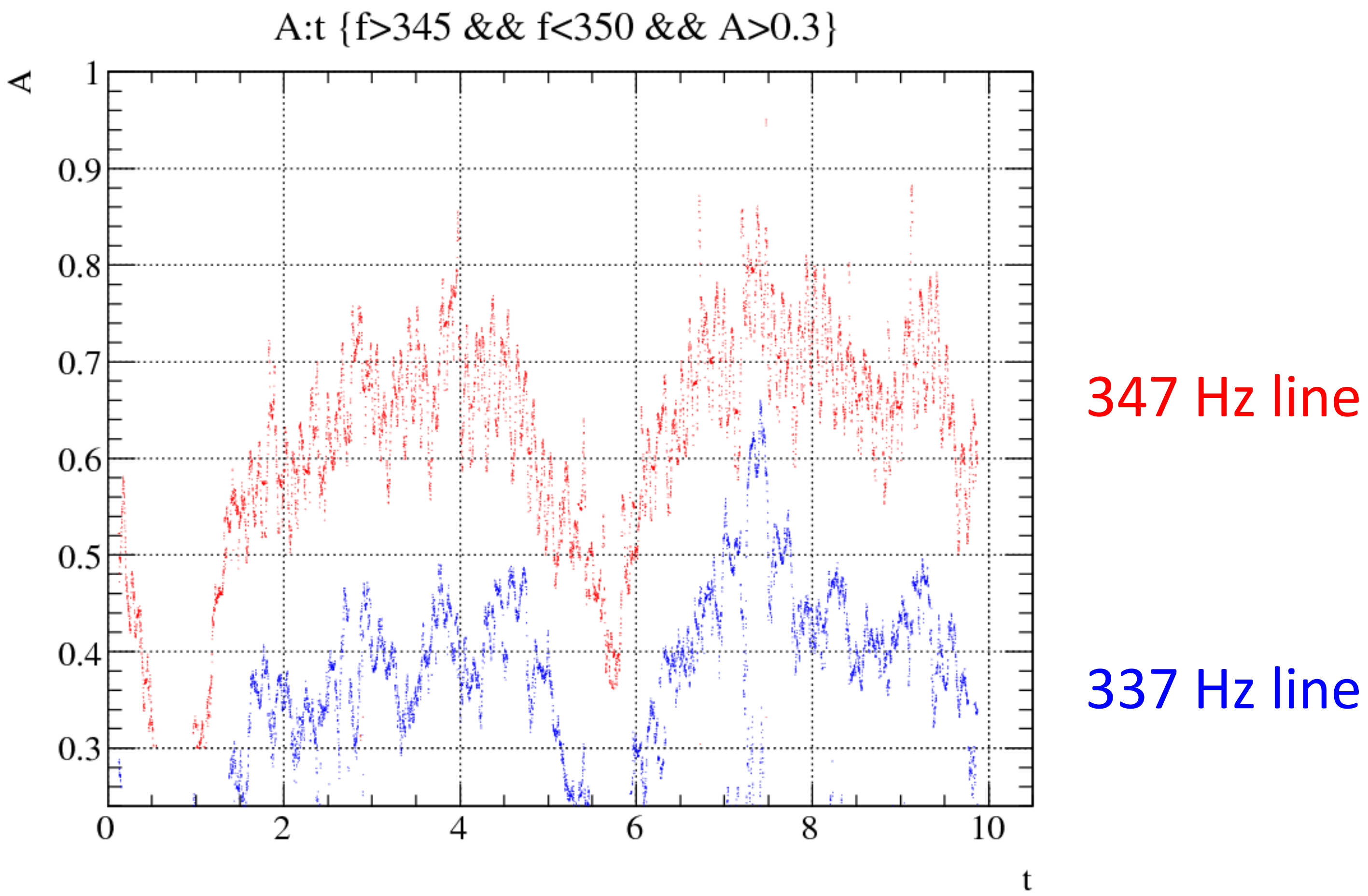

With NHA, you can not only know the line amplitude and frequency, but also their detail temporal behavior.

# 256 – 512 Hz

# fs=1024Hz; frame=256; shift=1; #spectrum=20



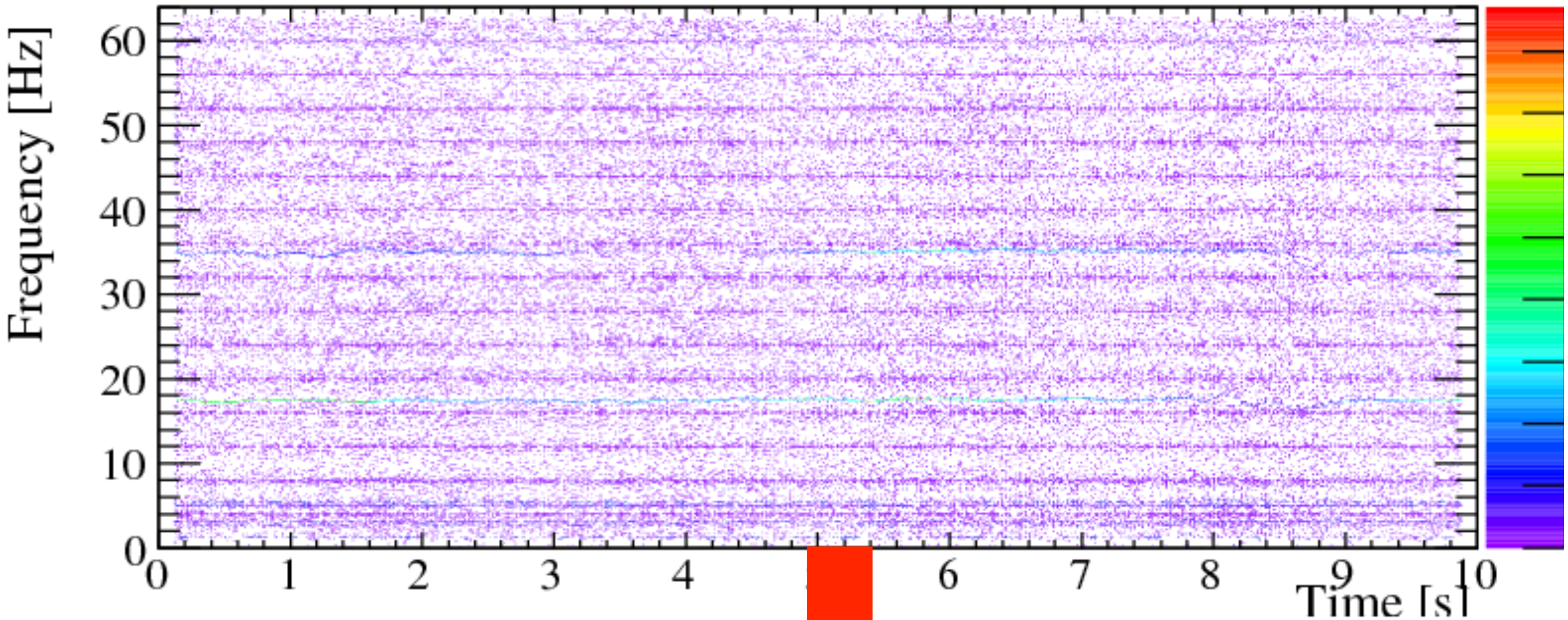
# Time variation of amplitudes of 337 and 347 Hz lines



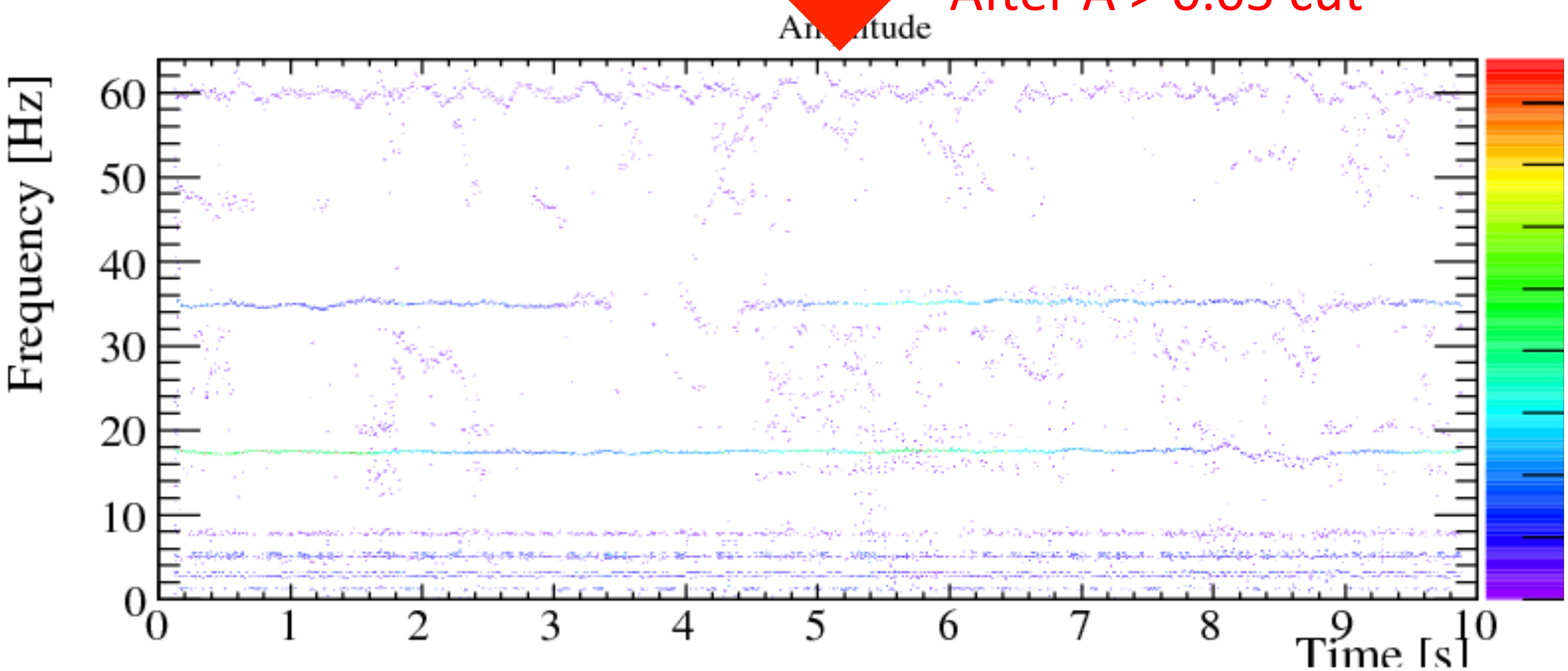

You can see a clear correlation of time variation of amplitude of the two lines.

< 64 Hz

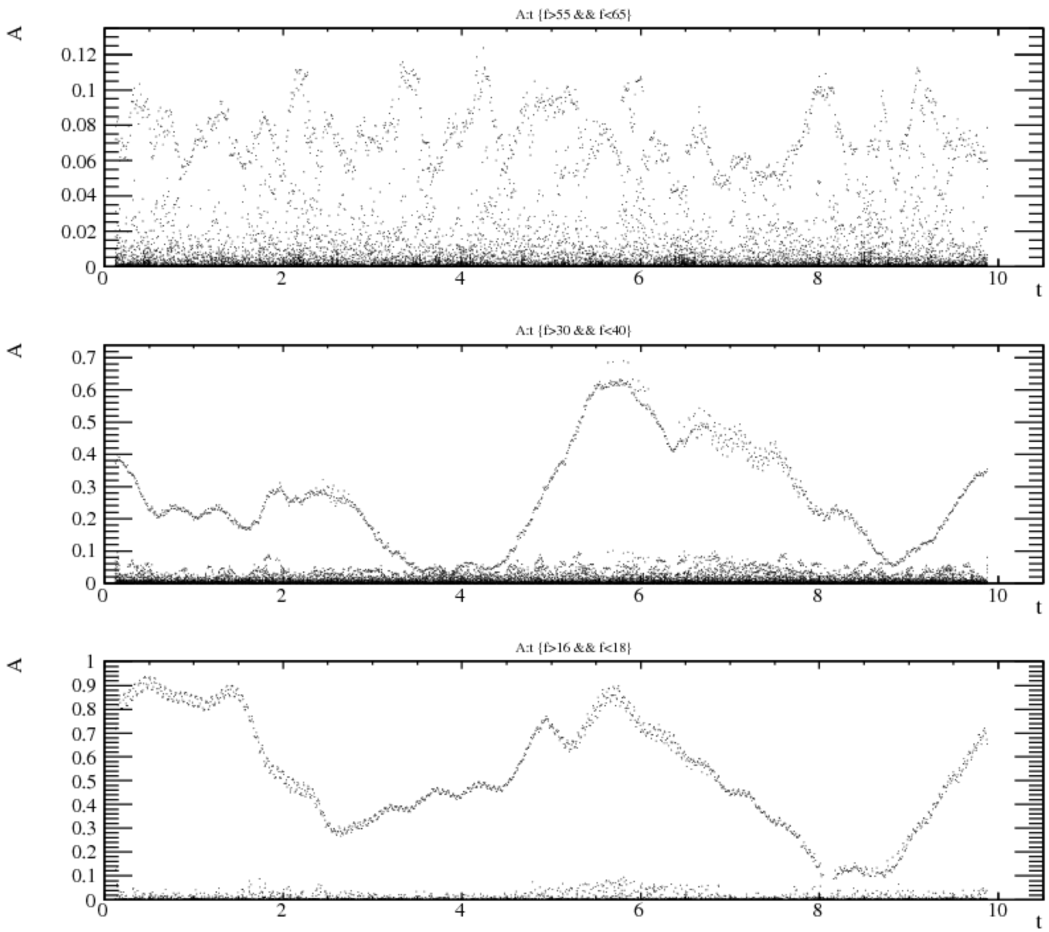
fs=128Hz; frame=32; shift=1; #spectrum=100




After A > 0.05 cut



# Time variation of amplitudes of lines below 64 Hz




60 Hz line

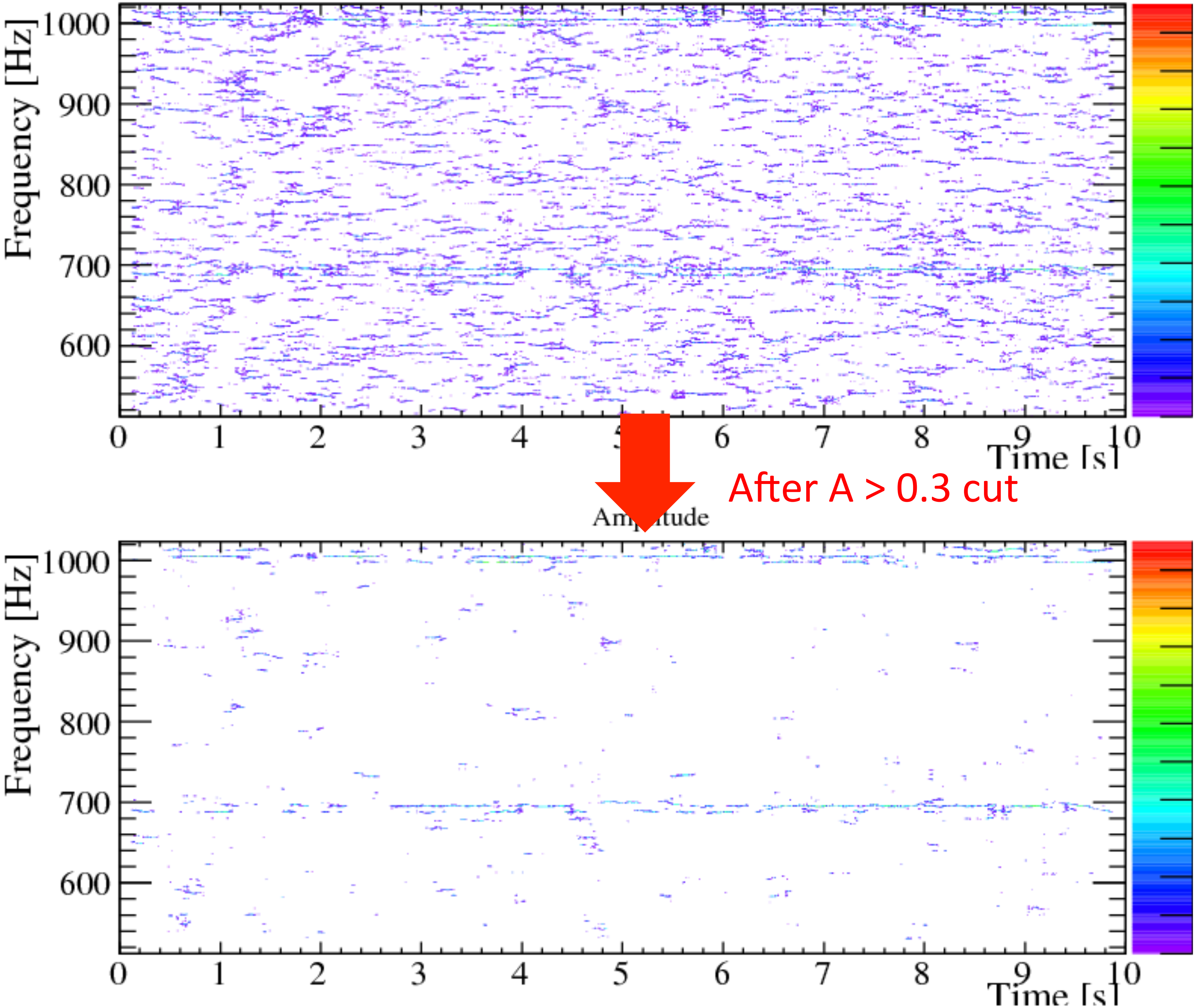
35 Hz line

17 Hz line

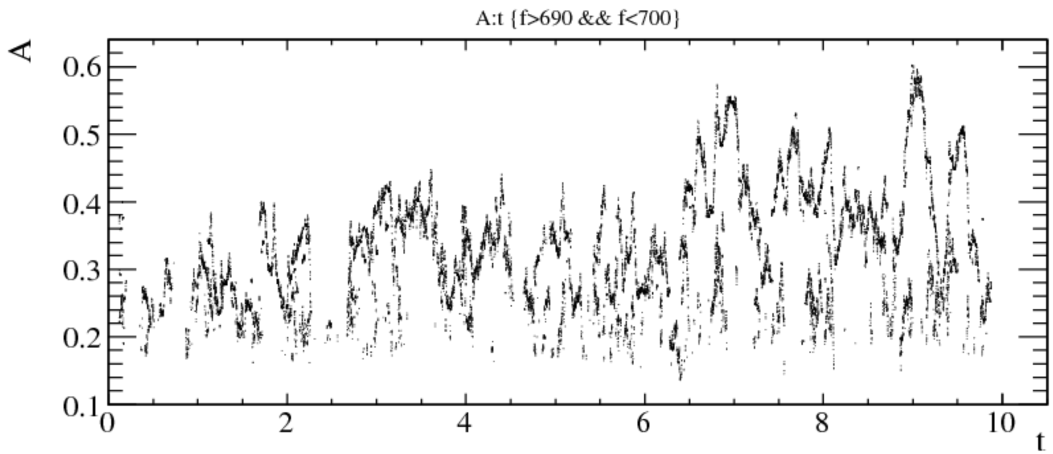
You can see the line feature with a resolution of << 1 sec.

512 - 1024 Hz

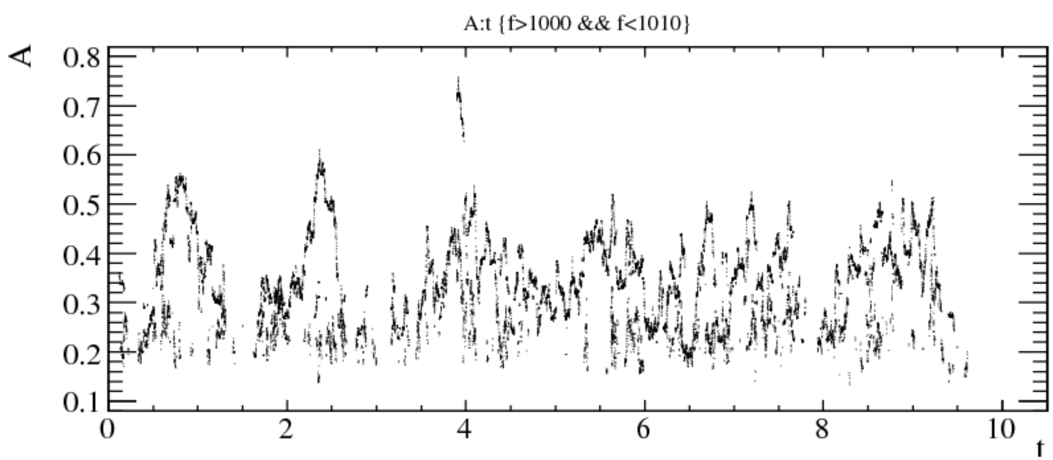
fs=2048Hz; frame=512; shift=1; #spectrum=20



# Time variation of amplitudes of 695 and 1005 Hz lines




695 Hz line




1005 Hz line

You can see the line feature with a resolution of << 1 sec.

# Summary

- Based on a paper on NHA, I implemented a simple C code of Iterative Least Square (ILS).
- Performance of this code would be worse than the full-fledged NHA, but it is C-based and easy for everybody to use (and also easy to wrap in Haskell)
- It could reveal the short-time time variation of 350 Hz lines amplitudes, which may give some hints to characterize lines and other possible glitches.

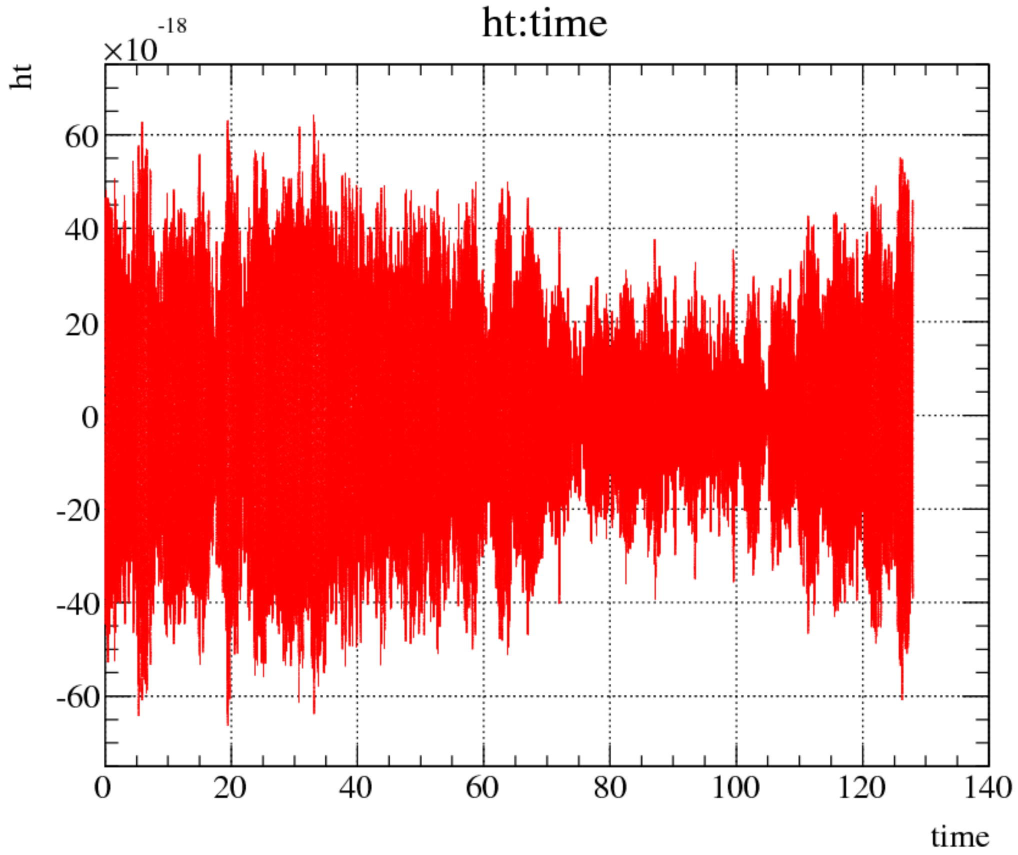
- If you have any data (including simulation) whose detail time structure you need to know, please feel free to ask me to analyze it! (though I cannot guarantee 100% quality)

# Backups

# Similar analysis for LIGO S6 data

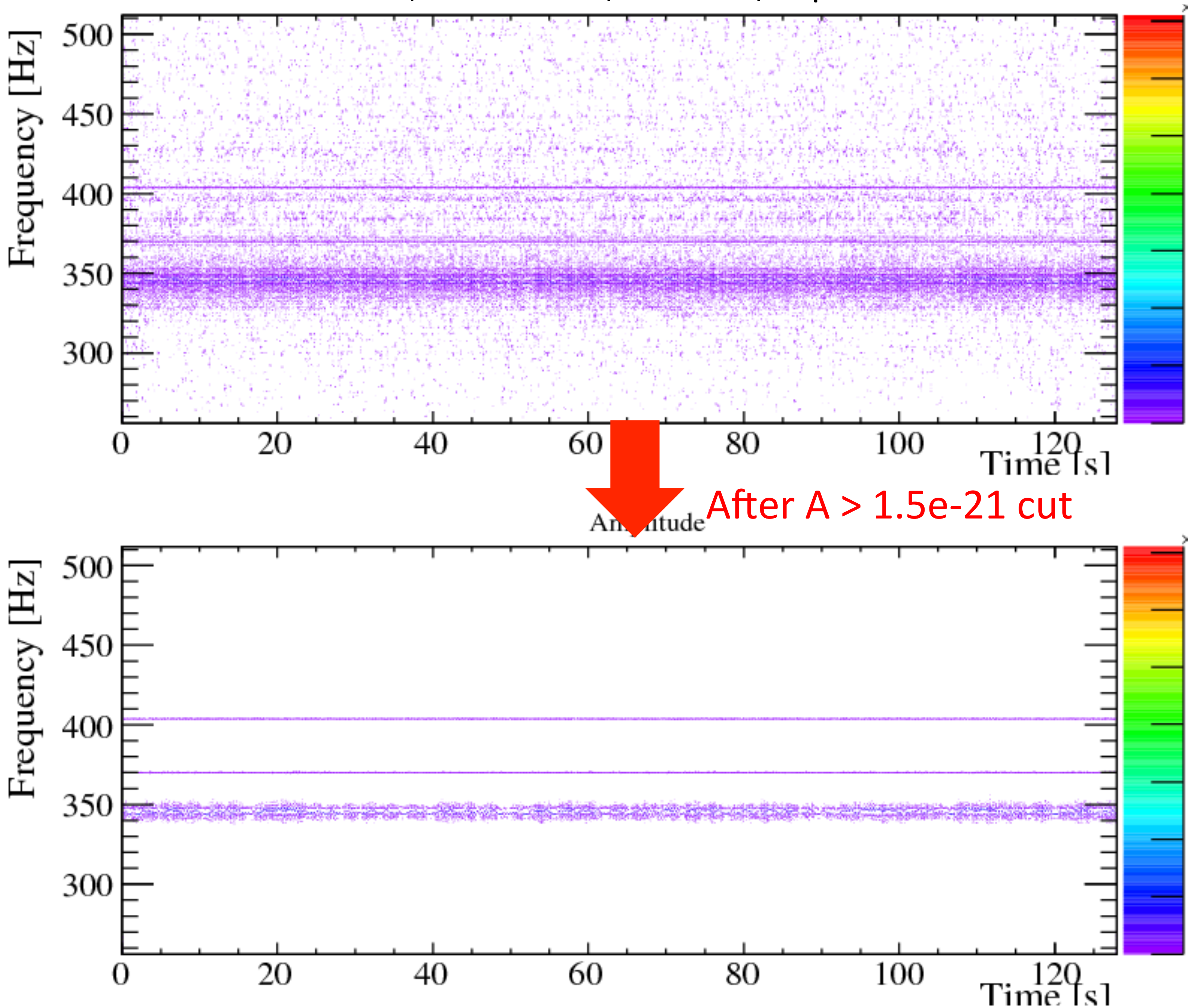
# S6 data

- I analyzed LIGO S6 data
  *L-L1_LDAS_C02_L2-959200000-128.gwf*
  - T = 128 s
  - fs = 16384 Hz

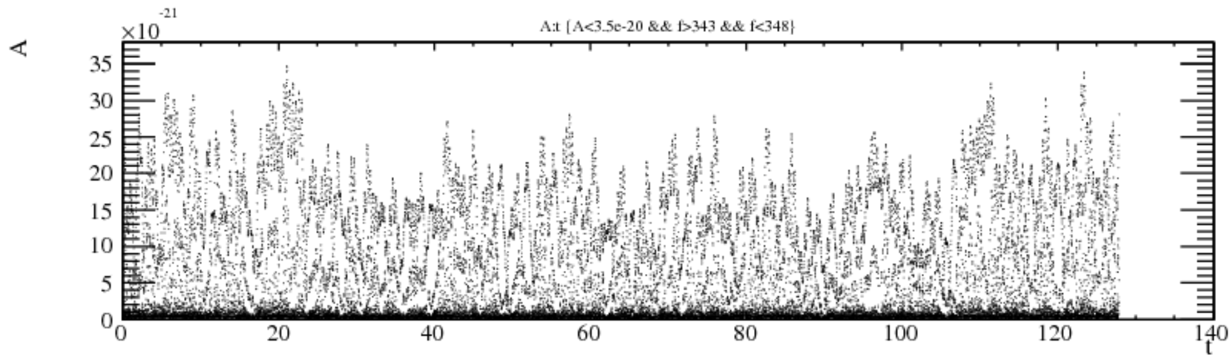


# 256 – 512 Hz

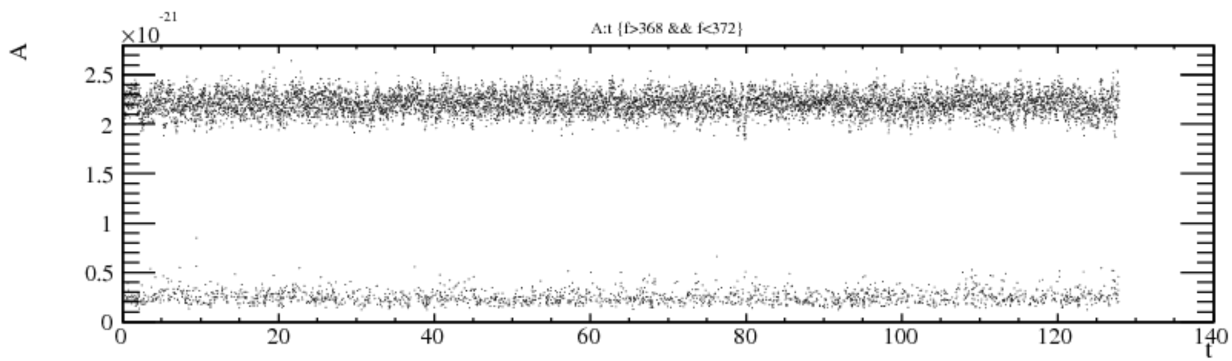
fs=1024Hz; frame=256; shift=16; #spectrum=20



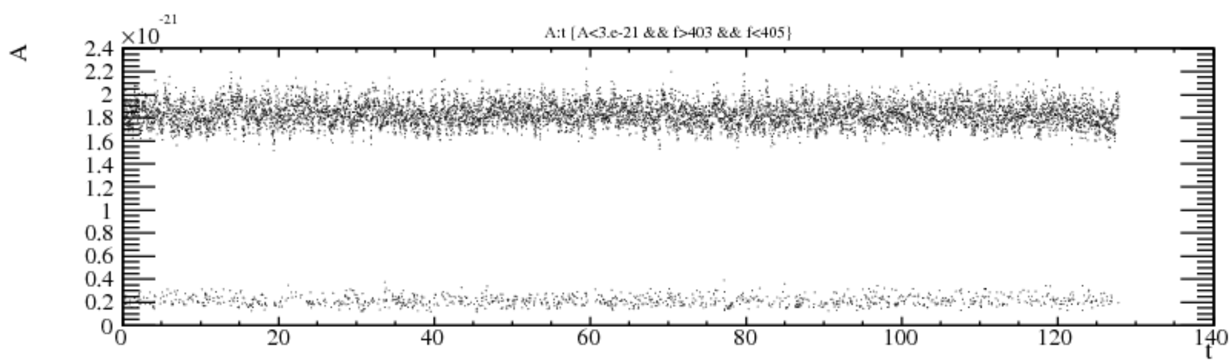
# Time variation of amplitudes of 345 and 370 Hz lines




345 Hz line



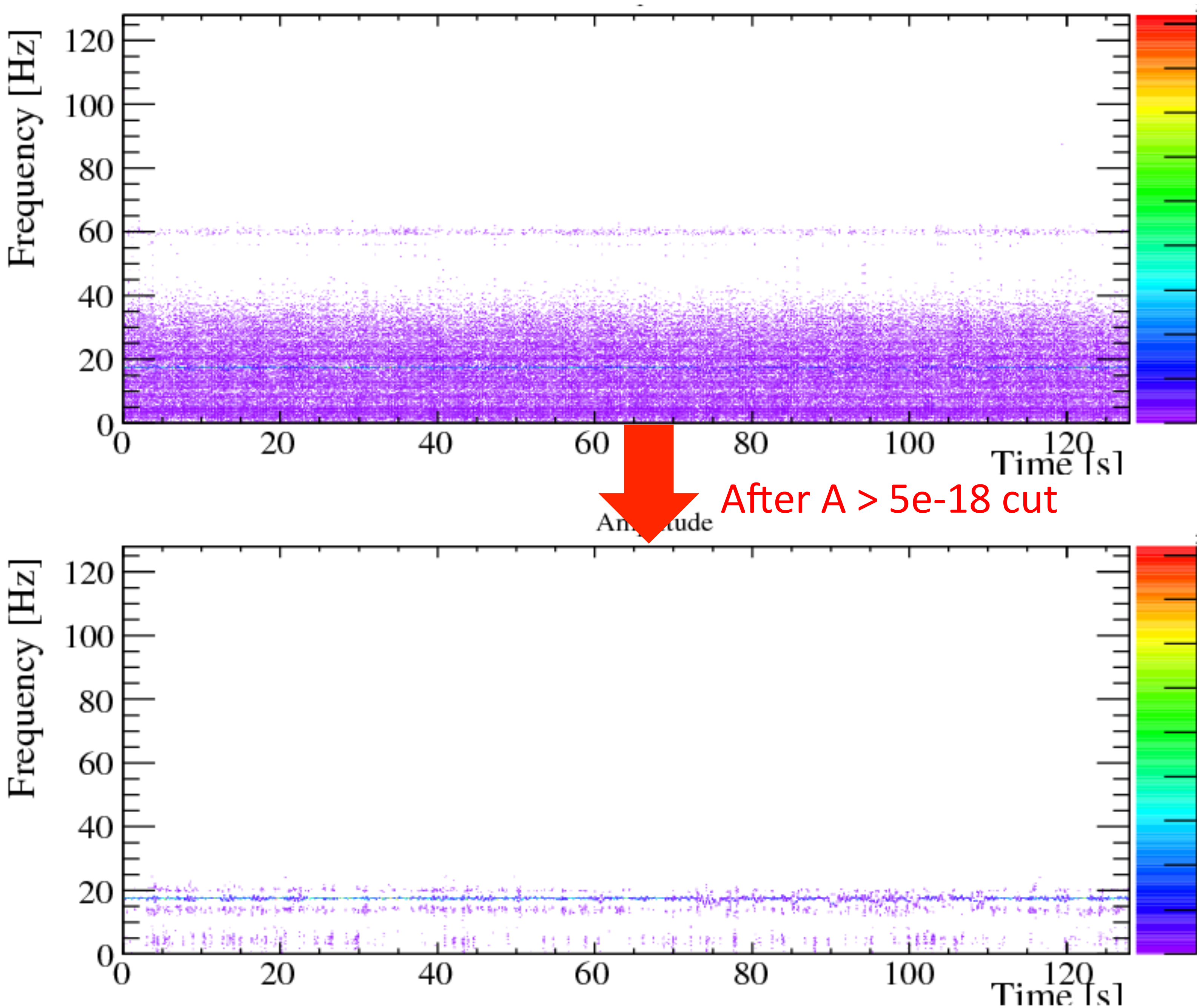

370 Hz line



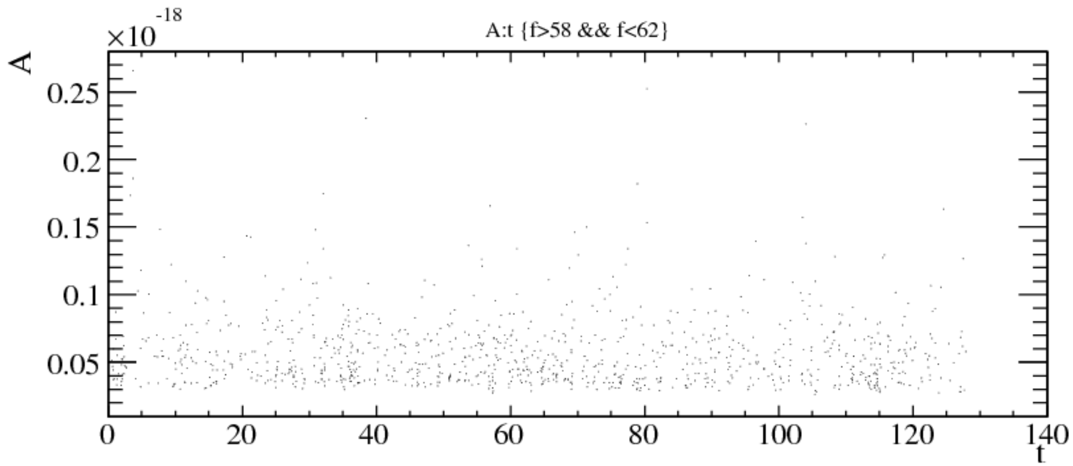

403 Hz line

< 128 Hz

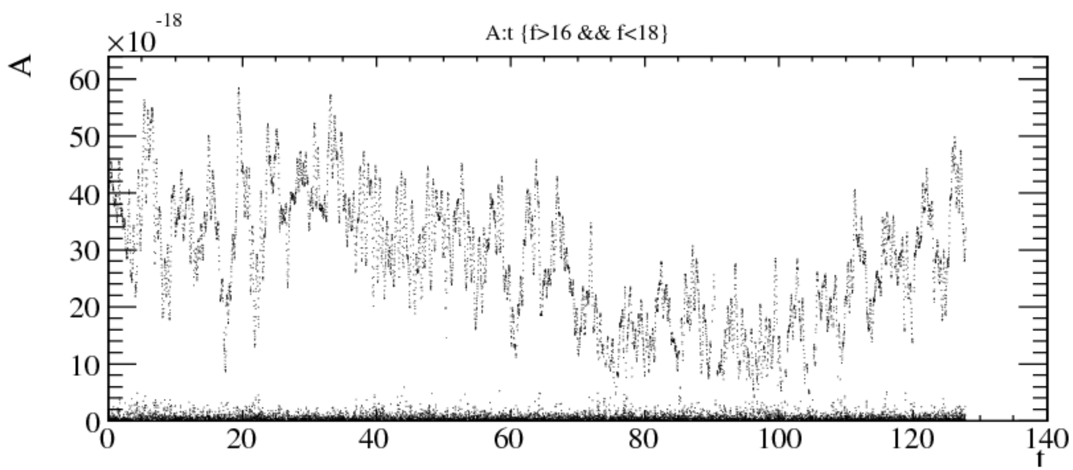
fs=256Hz; frame=64; shift=4; #spectrum=20



# Time variation of amplitudes of 35 and 60 Hz lines




60 Hz line




17 Hz line

The large amplitude variation of the 17 Hz line is found to cause that of the whole 128 s time-series data.