Burst Analysis in Wavelet Domain for multiple interferometers

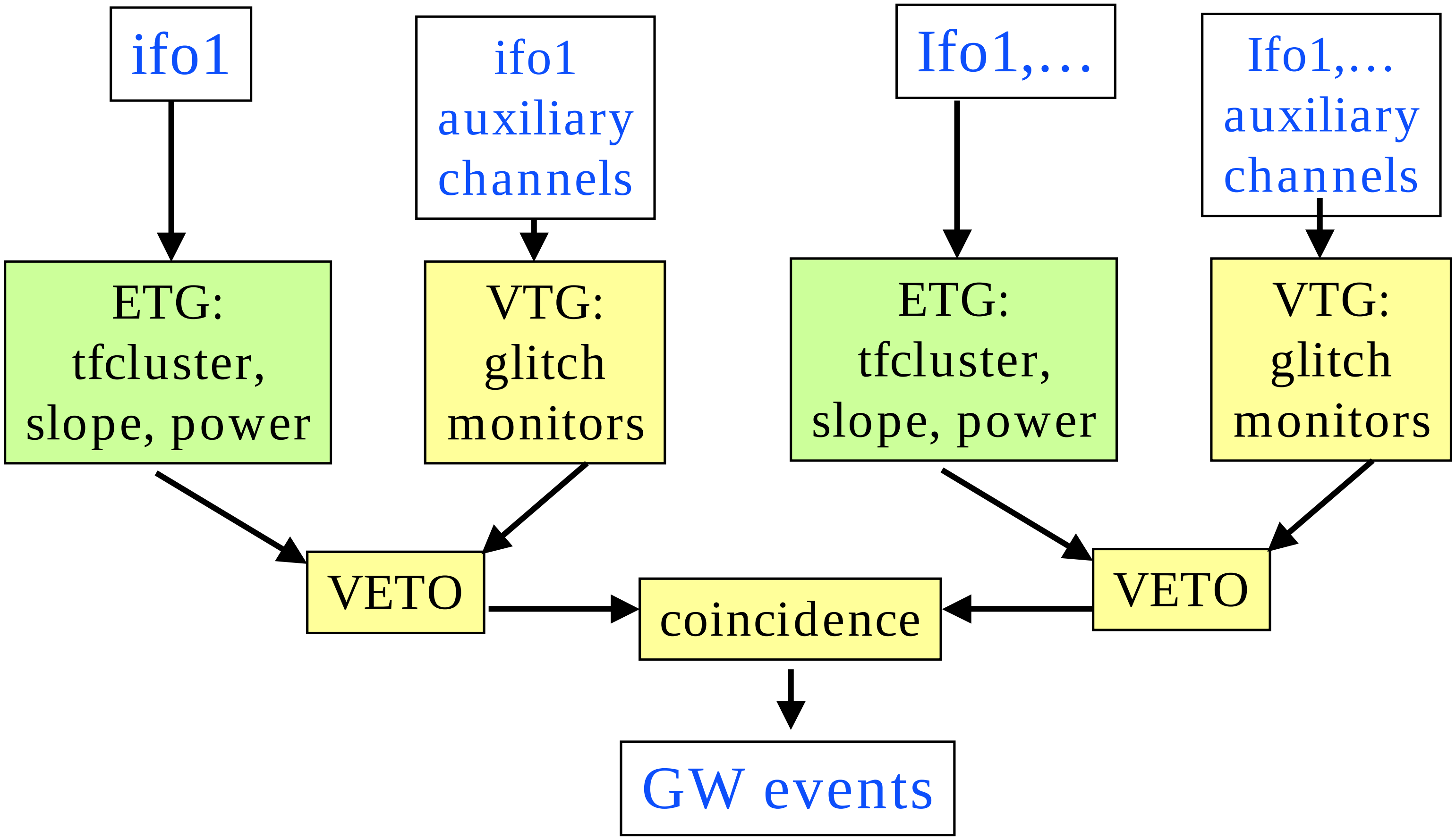
LIGO-G030132-00-Z

Sergey Klimenko

University of Florida

- **Analysis strategy**
- **Data analysis pipeline**
  - **WaveMon veto generator (S.Klimenko, see LIGO note G020383-00-Z)**
  - **WaveBurst event generator (S.Klimenko, I.Yakushin)**
- **Analysis algorithms**
- **Some results for S1**
- **Summary & Plans**

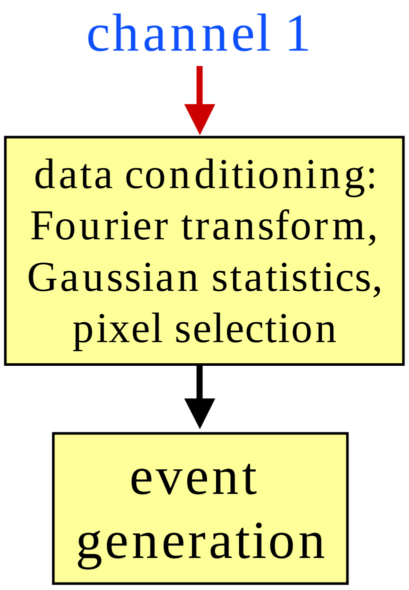
Analysis strategy



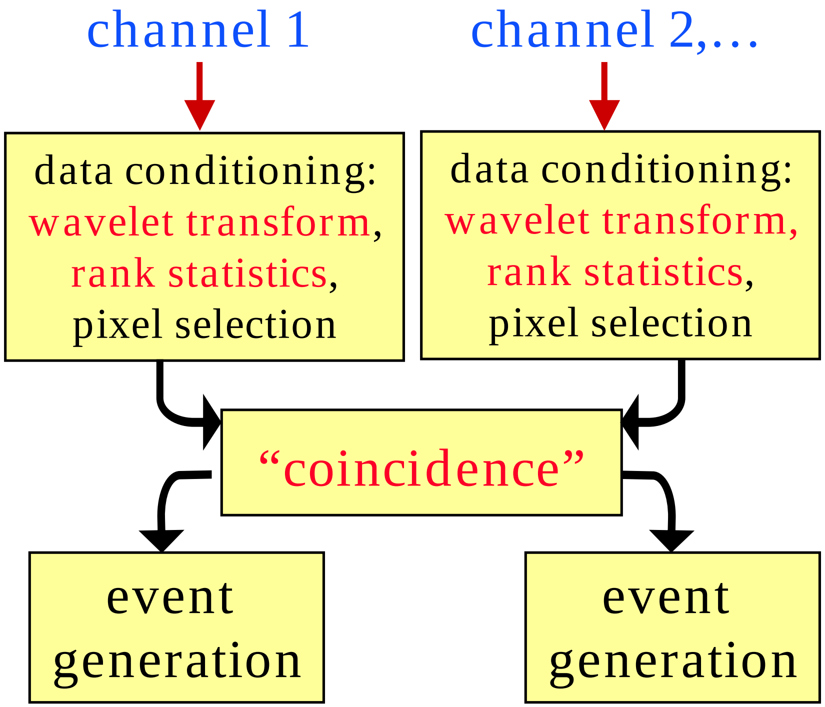
- **Triggers are generated independent for each IFO**
- **GW candidates are generated on the final coincidence step**

event trigger generators

TFCLUSTER

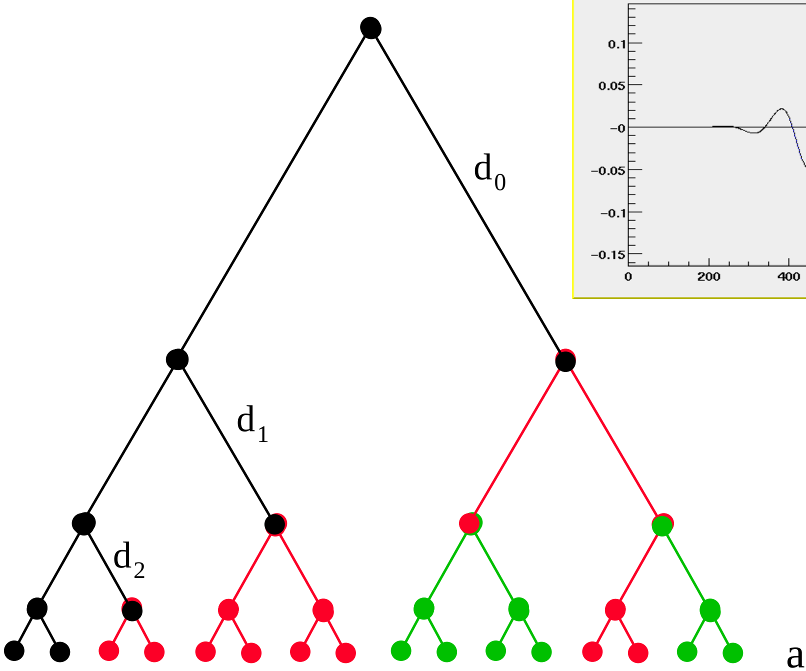


WaveBurst / WaveMon

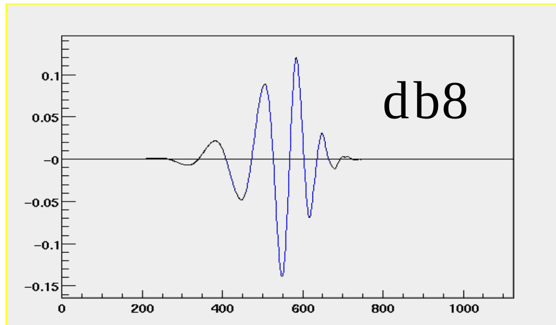


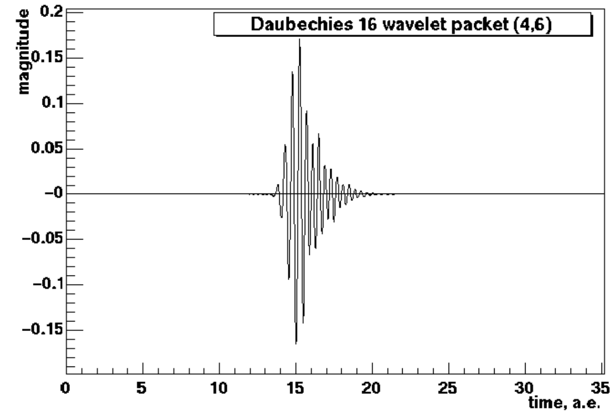
- **pixels produced by artifacts are clustering together - very powerful idea behind the Time-Frequency cluster analysis**
- **use coincidence to enhances it.** $$(powerful\_idea)^2$$
- **coincidence is used before the events are generated → approach often used in HEP.**

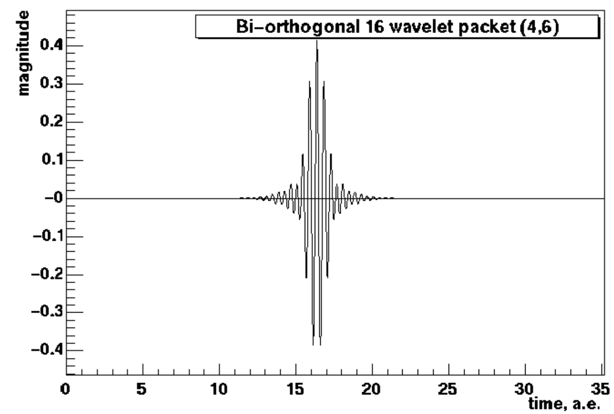
Wavelet Transform



wavelet transform tree

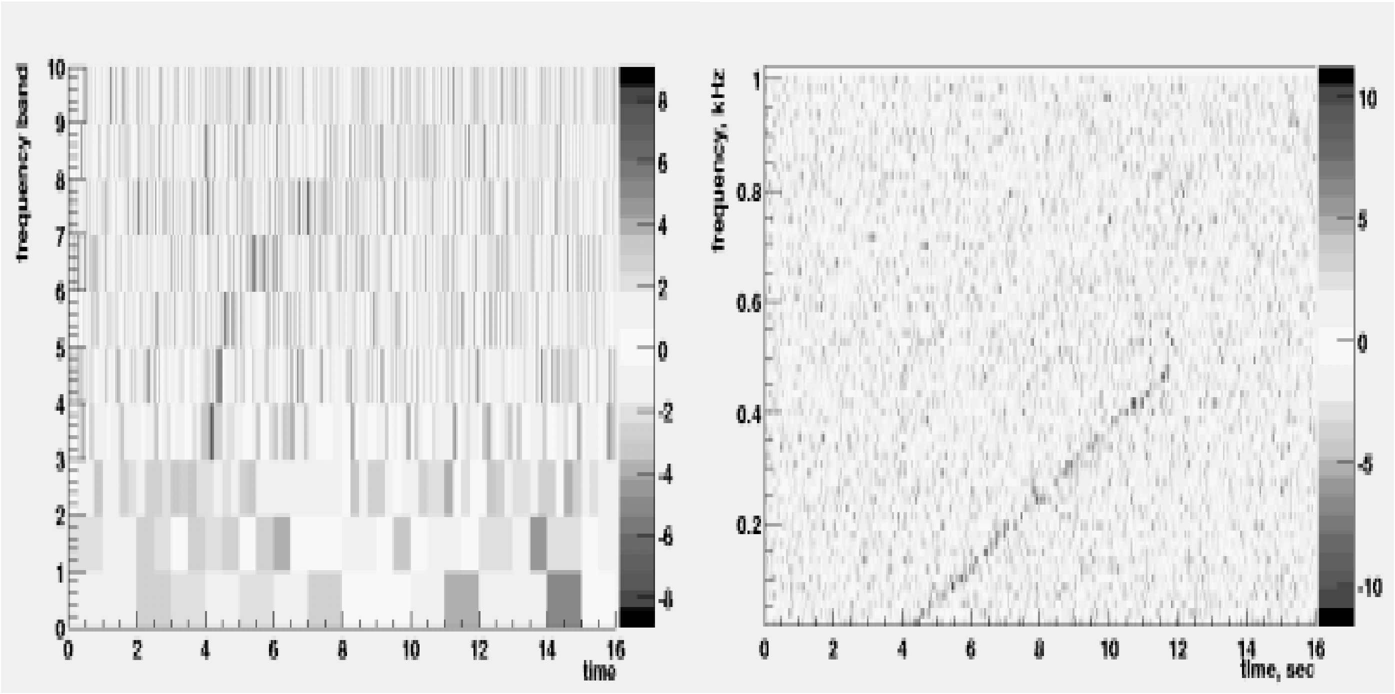






- Optimize search by selecting wavelet type & decomposition tree
- detail coefficients d_i represent data in different frequency bands
  - $df = f/2, f/4, f/8, \ldots$ - dyadic, $df = f/n$, - linear basis

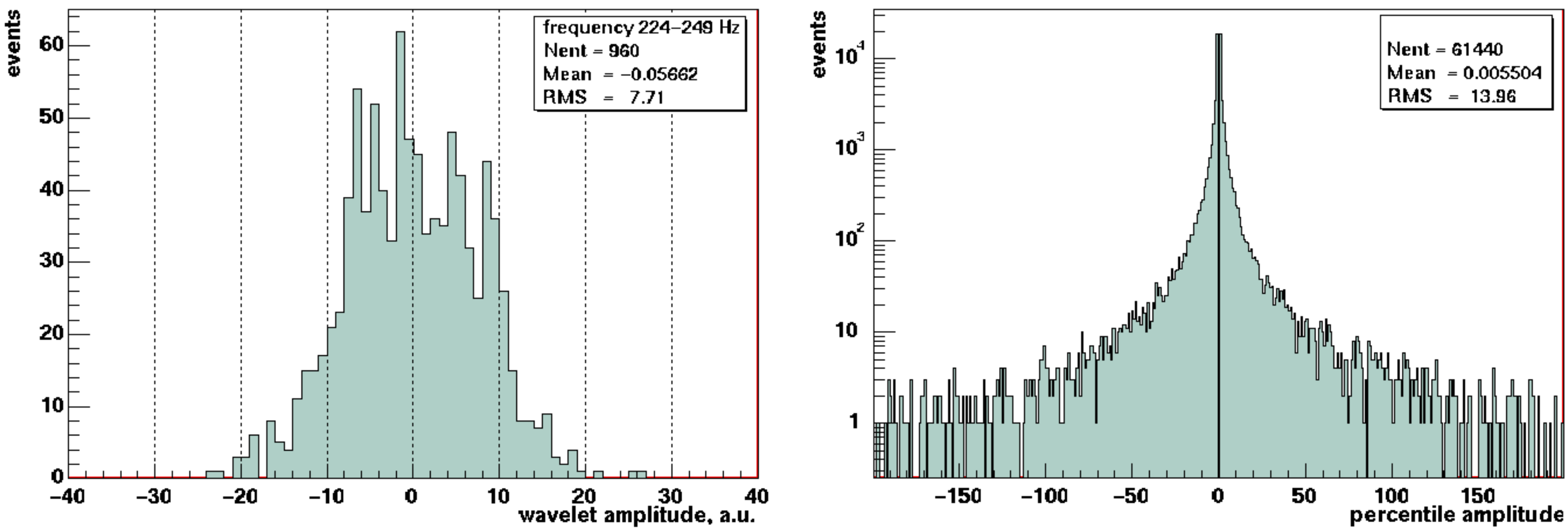
wavelet time-scale(frequency) plane



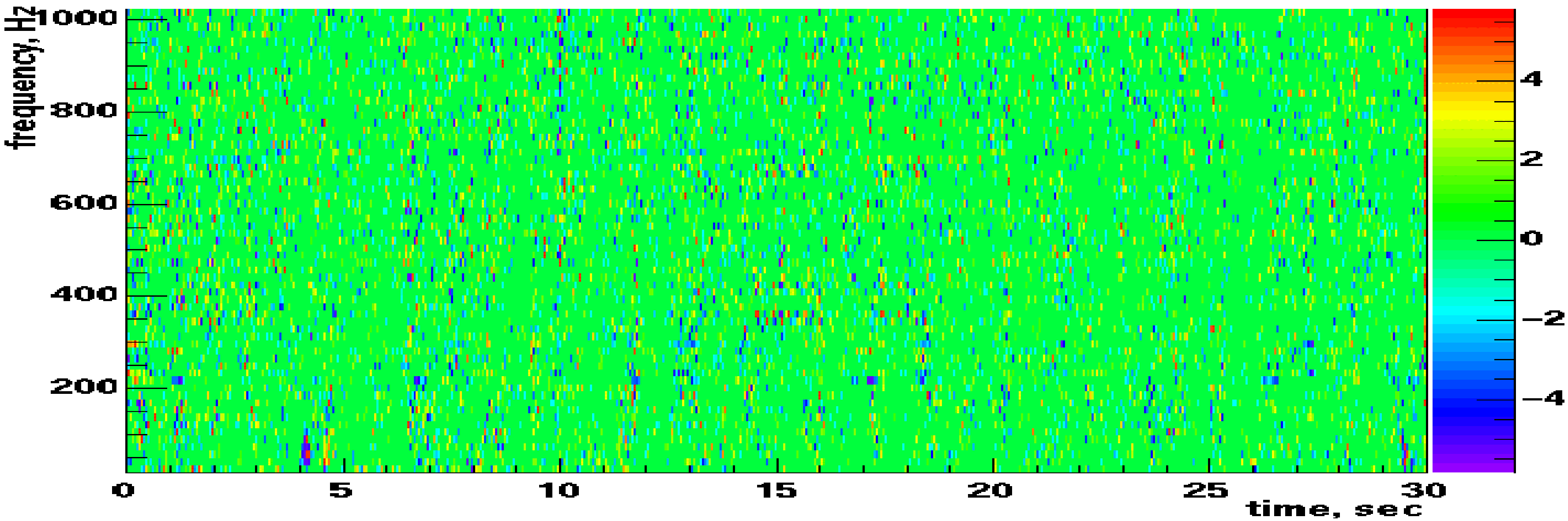
- For short (therefore broadband) bursts dyadic WT should be good.
- For long signals, like inspirals, binary WT could be optimal
- Several wavelet transforms are implemented in the datacondAPI
- Currently used in WaveBurst:
  - interpolating bi-orthogonal, Daubechies, Simlets
  - binary wavelet tree (linear frequency scale)

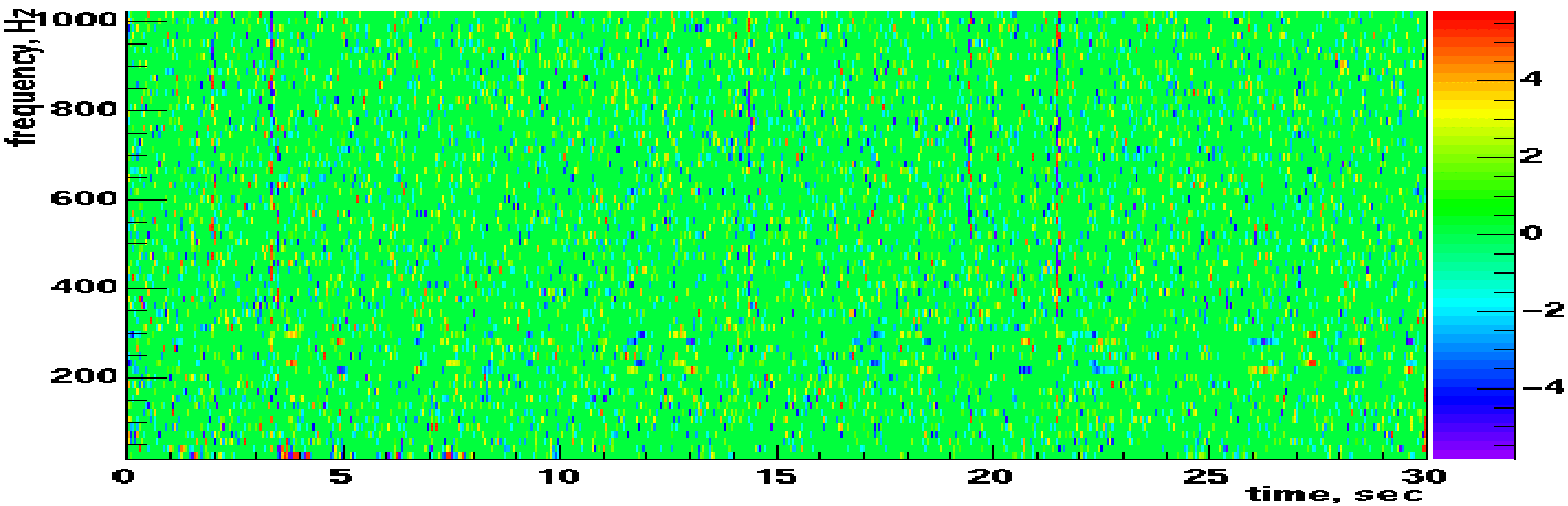
Percentile Transform

- given a sample with amplitude *A*, define its *percentile amplitude* as *a=1/f*, where *f* is fraction of samples in the population with absolute value of amplitude greater then |*A*|
- Equivalent to calculation of rank statistics → *non-parametric*
- percentile amplitude distribution function: $P(a)=1/a^2$
  - the same for all wavelet layers



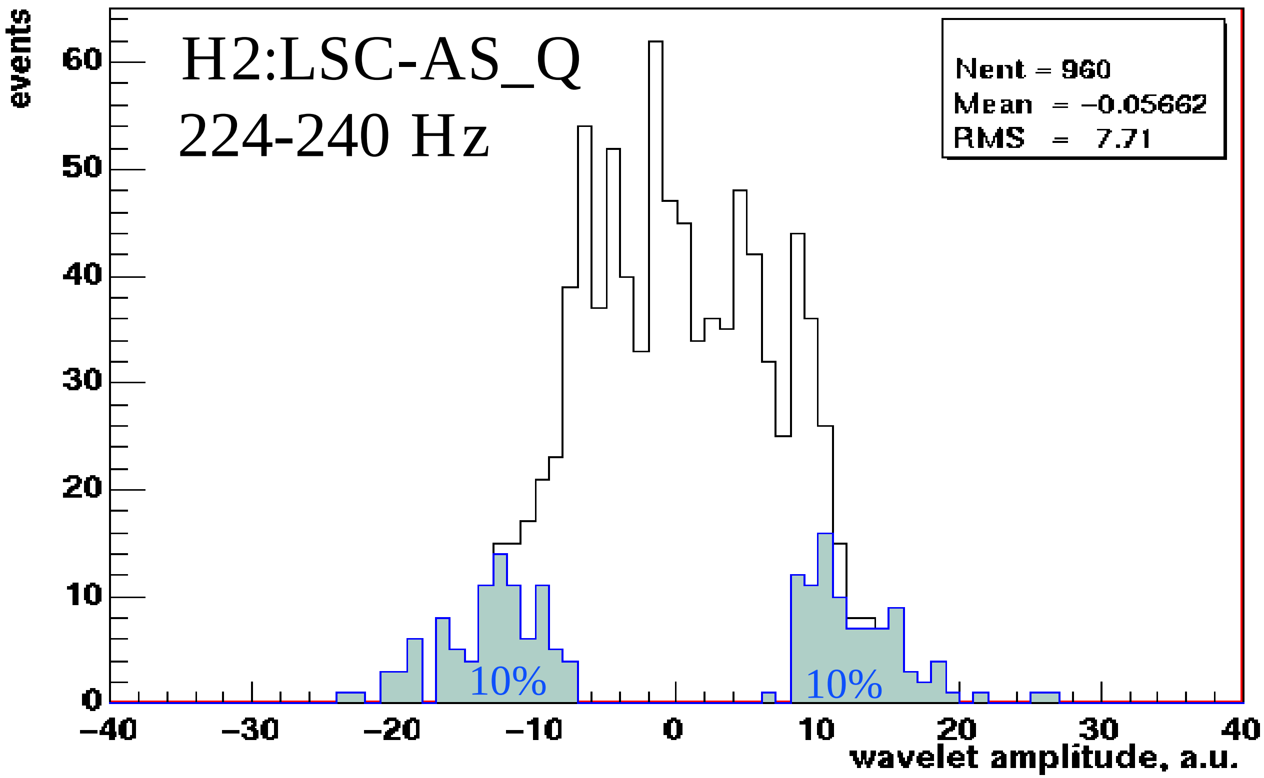
raw → percentile TF plots



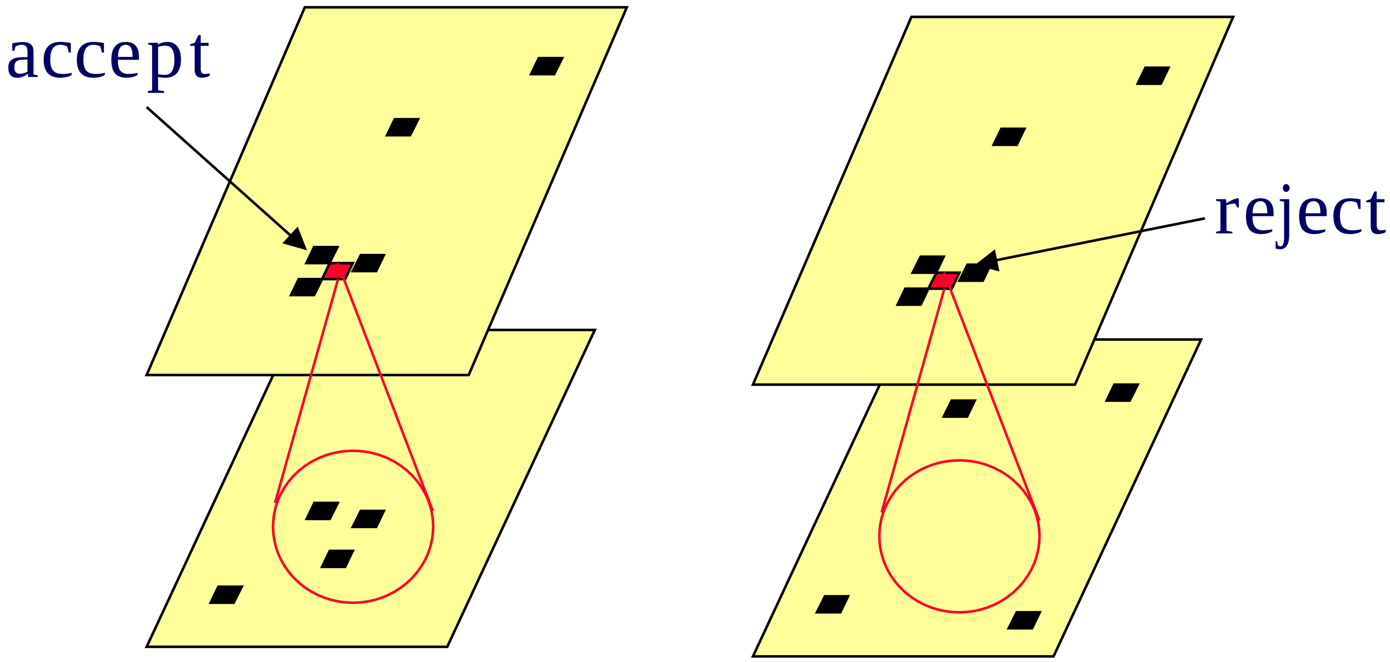


Percentile Thresholding Rule

- threshold on percentile amplitude
- select a certain fraction of samples (set T-F occupancy)
- don't care about the data distribution function
- one possible way of "de-noising of wavelet data"



Coincidence Rules



- **Given local occupancy $O_i(t,f)$ in each channel, after coincidence the cluster occupancy is**

$$O_C(t,f) = \alpha \cdot O_1(t,f) \cdot O_2(t,f)$$

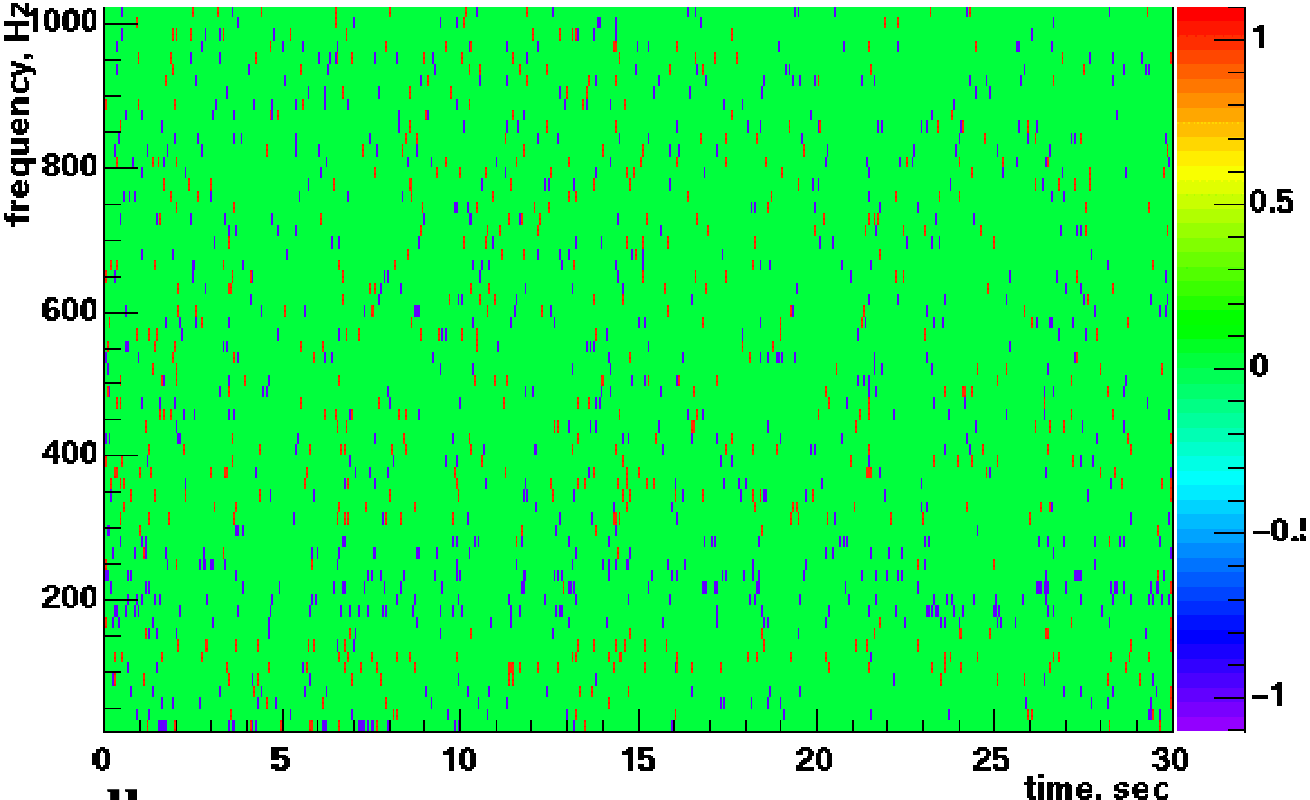
- **For strict coincidence $O_i(t,f) = p^2$ for random pixels produced by noise.**

  for example if p=0.1, average occupancy after coincidence is 0.01

- **WaveBurst can use various coincidence policies → allows customization of the pipeline for specific burst searches.**

strict coincidence

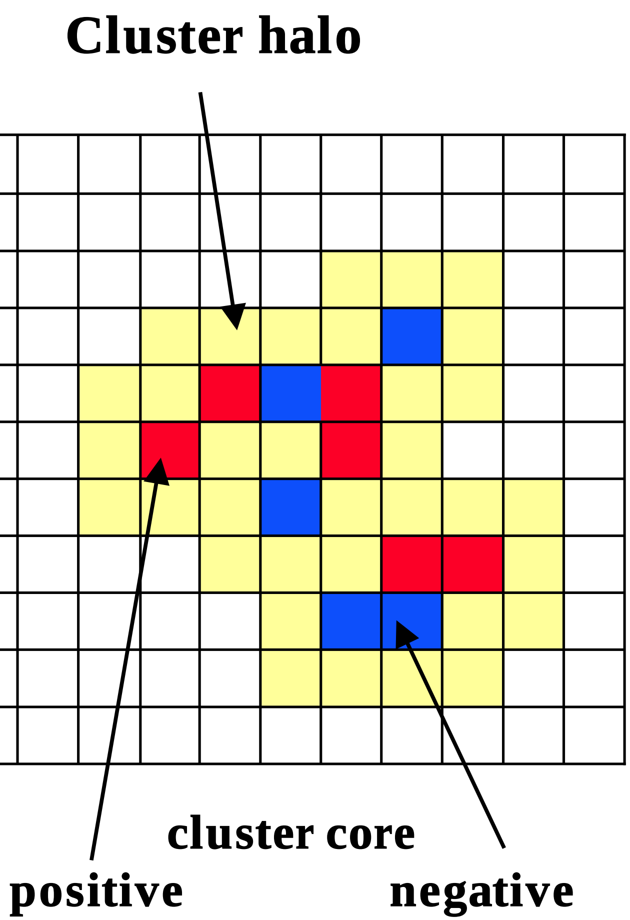
- coincidence TF plot for LSC-AS_Q && IOO-MC_F



- coincidence allows
  - set a low threshold at the previous (thresholding) step
  - "build" clusters with known false alarm rate, determined by random clustering of pixels

Cluster Analysis

cluster – T-F plot area with high occupancy



Cluster Parameters

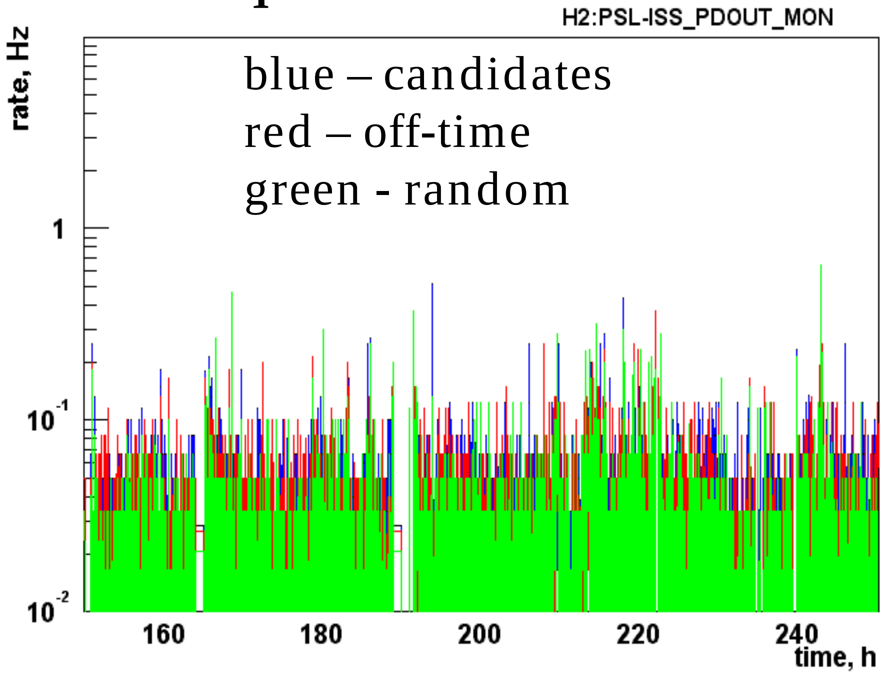
size – number of pixels in the core
volume – total number of pixels
density – size / volume
amplitude – maximum amplitude
power - wavelet amplitude / w50
energy - power x size
asymmetry – (#positive - #negative) / size
likelihood – $\log(\Pi A_i)$
neighbors – total number of neighbors
frequency - core minimal frequency [Hz]
band - frequency band of the core [Hz]
time - GPS time of the core beginning
interval - core duration in time [sec]

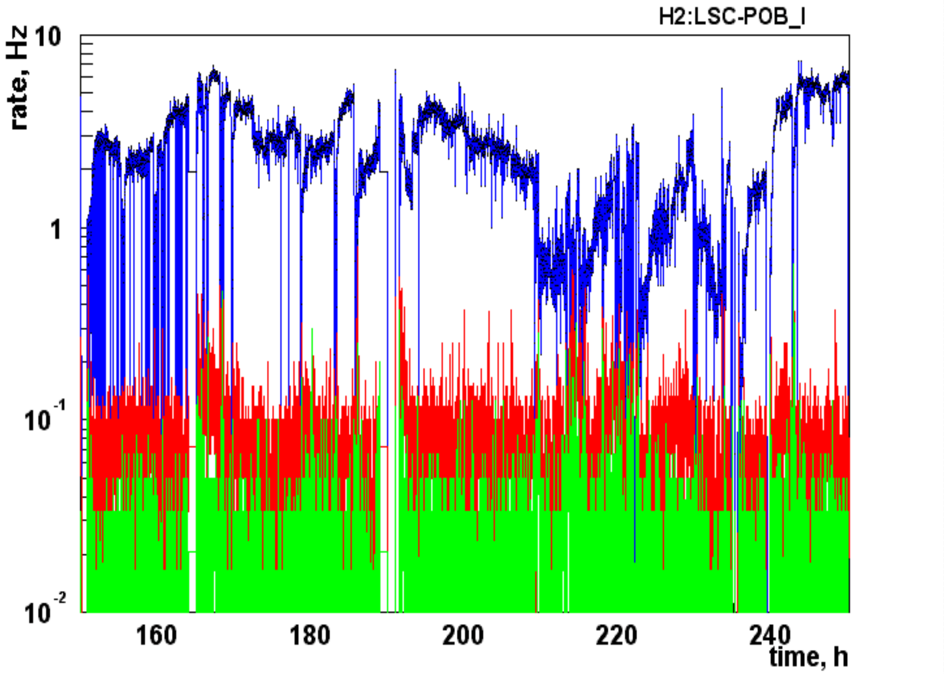
‘orthogonal’ data sets

- WaveMon/WaveBurst generate clusters of three types:
  - candidate clusters → potential GW bursts
  - off-time clusters (due to random coincidence of glitches) → veto tuning
  - “false alarm” clusters (due to random clustering of noise) → threshold settings

WaveMon rates for coincidence with H2:LSC-AS_Q

quiet channel S1 data hot channel

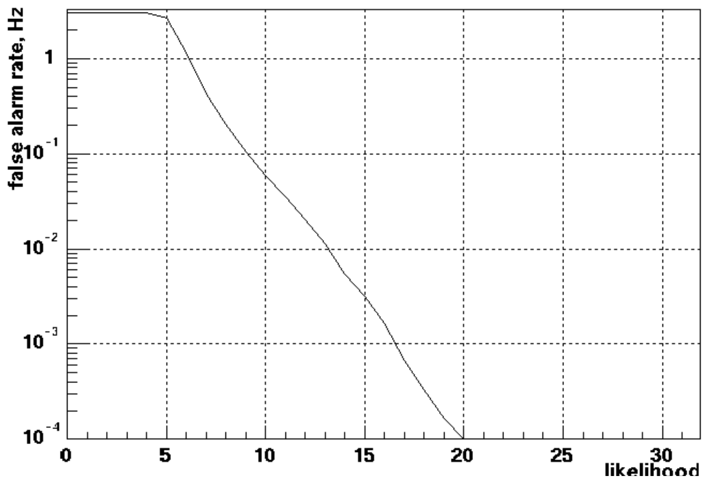


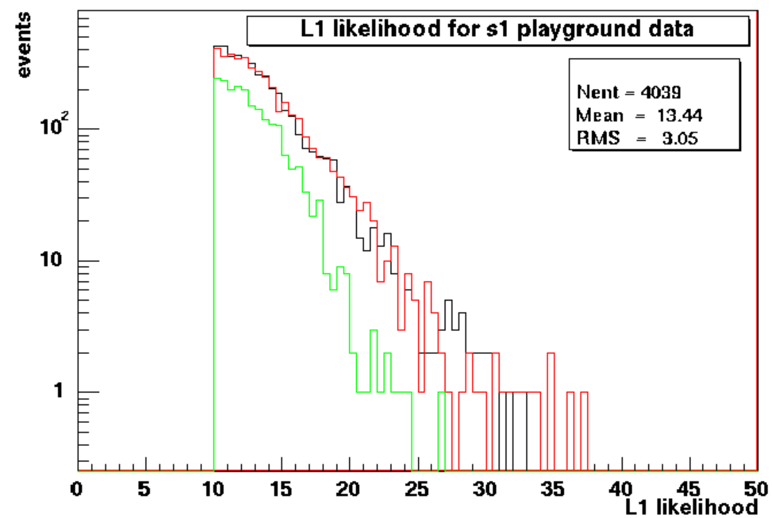


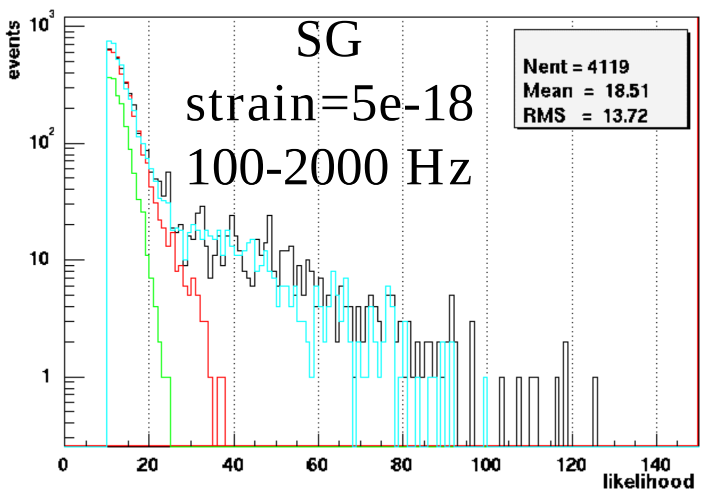
selection cuts: cluster likelihood

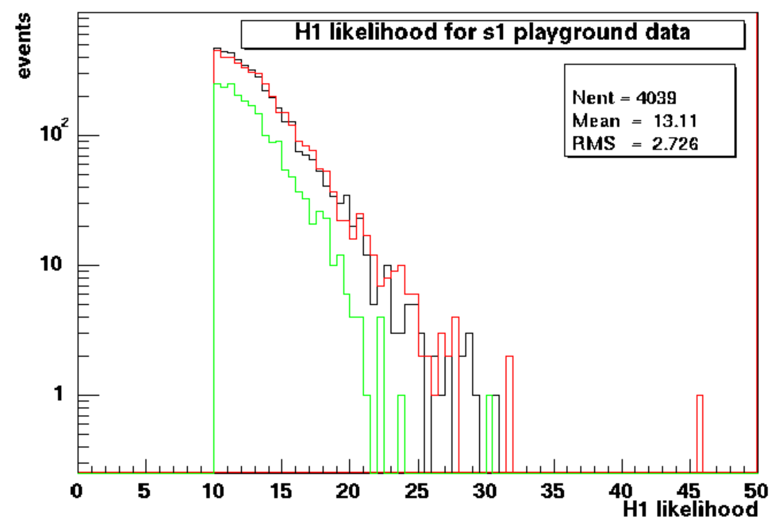
$$L = \log\left(\prod A_i\right)$$

- L threshold can be set a priori based on the false alarm rate
  - does not need tuning → perfect for real time processing
  - Only L threshold is applied at run time → ETG output rate ~0.2Hz







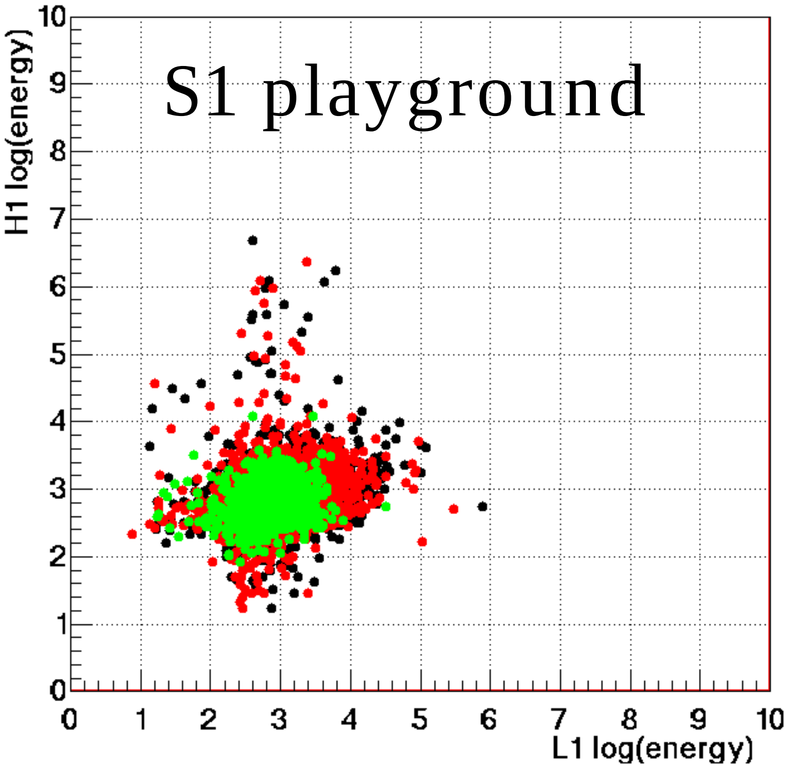


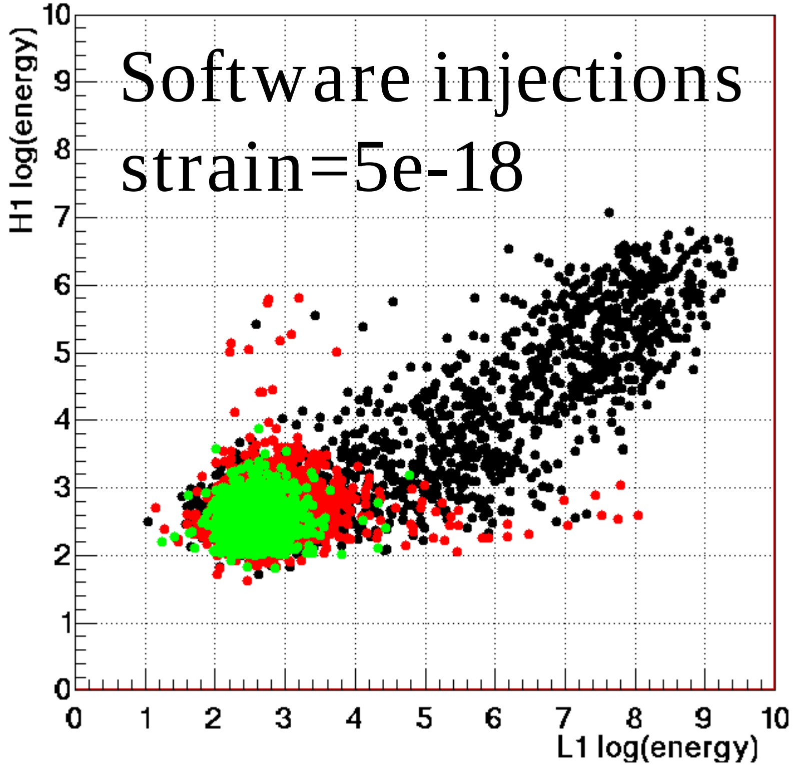
selection cuts: cluster "energy"

$$E = \sum \left(\frac{w_i - w_0}{0.67 w_{50}} \right)^2$$

w_i – wavelet amplitude
w_0 – distribution median
w_{50} – median of $|w_i - w_0|$

- **For Gaussian noise** $E=\chi 2$
- Energy cut is set based on the false alarm rate and it is used for post-processing event selection

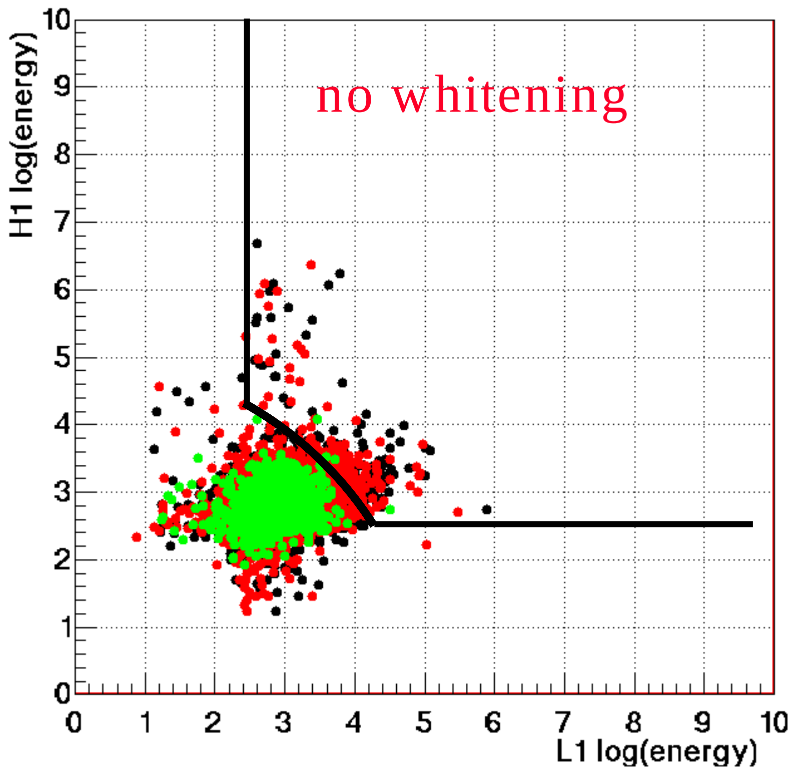


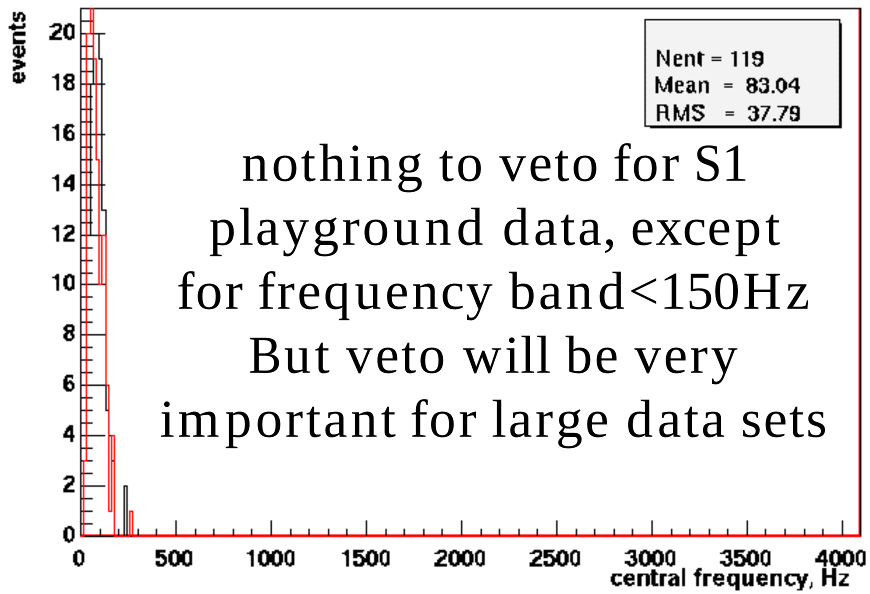


VETO

- **After applying selection cut, remaining events are suppose to be removed with vetoes**
- **Veto events can be generated by DMT monitors and by ETGs itself.**
- **Off-time data set (red dots) is used for selection of effective veto channels**

work in progress
on optimization of
veto strategy





Other issues

- **Simulation study using injected bursts:**
  - **pipeline consistency check.**
  - **efficiency study for different coincidence policies using bursts injected randomly in time and with random delay between detectors.**
- **X-correlation between burst candidates found in two detectors. Could be significant enhancement of the search pipeline. (see Malik's talk)**

Conclusion & Plans

- **WaveBurst ETG**
  - **produces GW burst candidates for a coincidence of two detectors**
  - **uses non-parametric statistics → allows *a priori* threshold settings for the on-line operation.**
  - **produce event triggers with known false alarm rate**
- **WaveMon monitors the detector glitch performance**
  - **Can help identify the source of glitches in the GW channel by looking at frequency content and correlation with auxiliary channels**
  - **produces veto triggers**
- **Plans**
  - **Use S1 as a playground for WaveBurst pipeline.**
  - **Process S2**
  - **Preparations for S3**