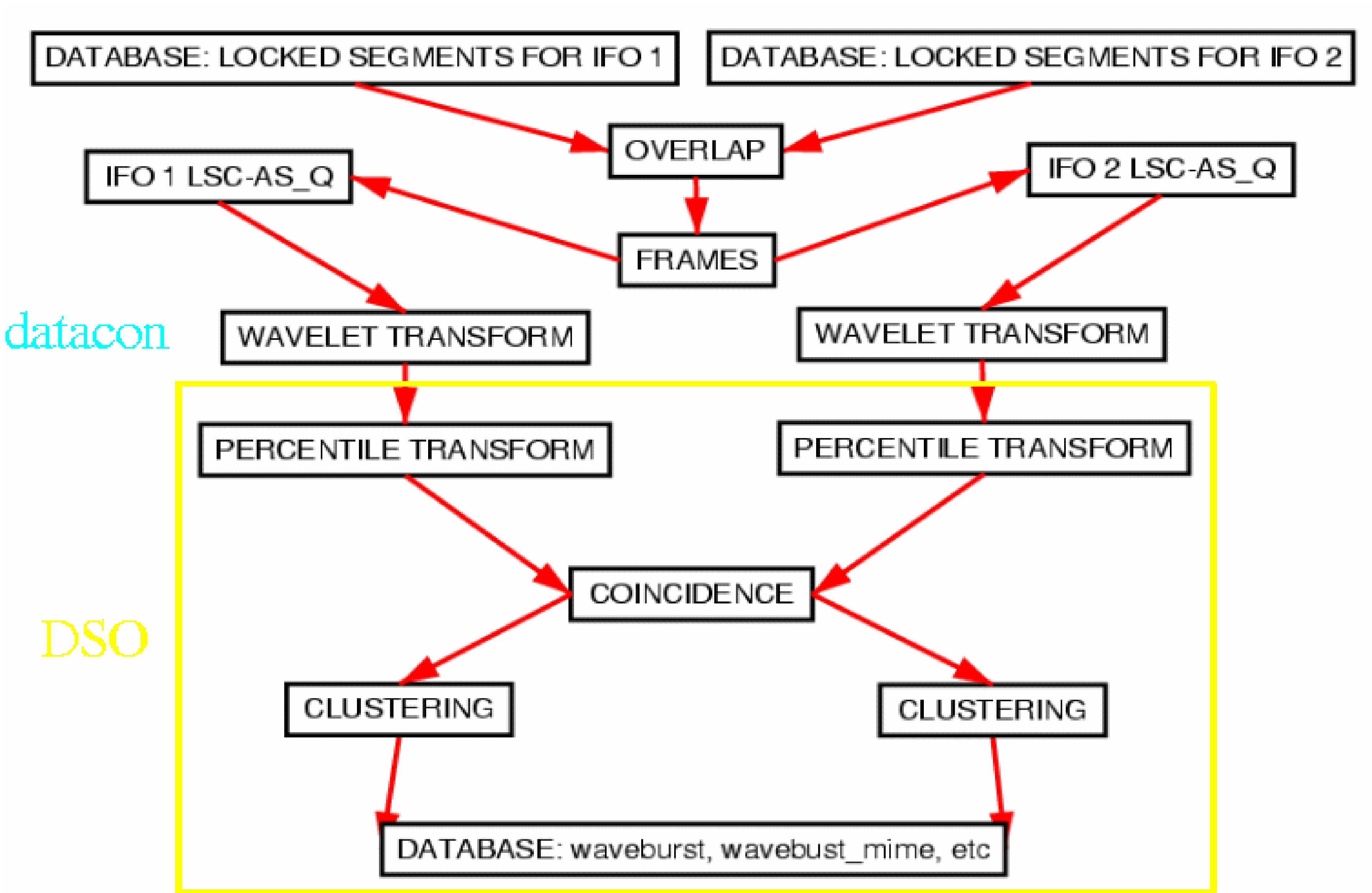
Waveburst DSO: current state, testing on S2 hardware burst injections

Sergei Klimenko
Igor Yakushin

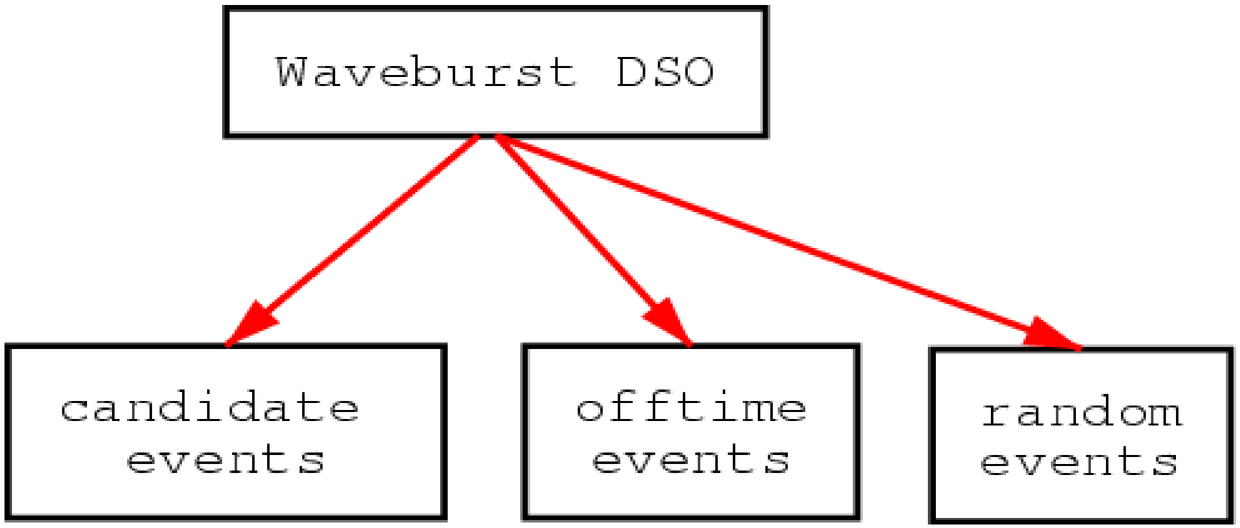
LSC meeting, March 2003

LIGO-G030101-00-Z

Data flow

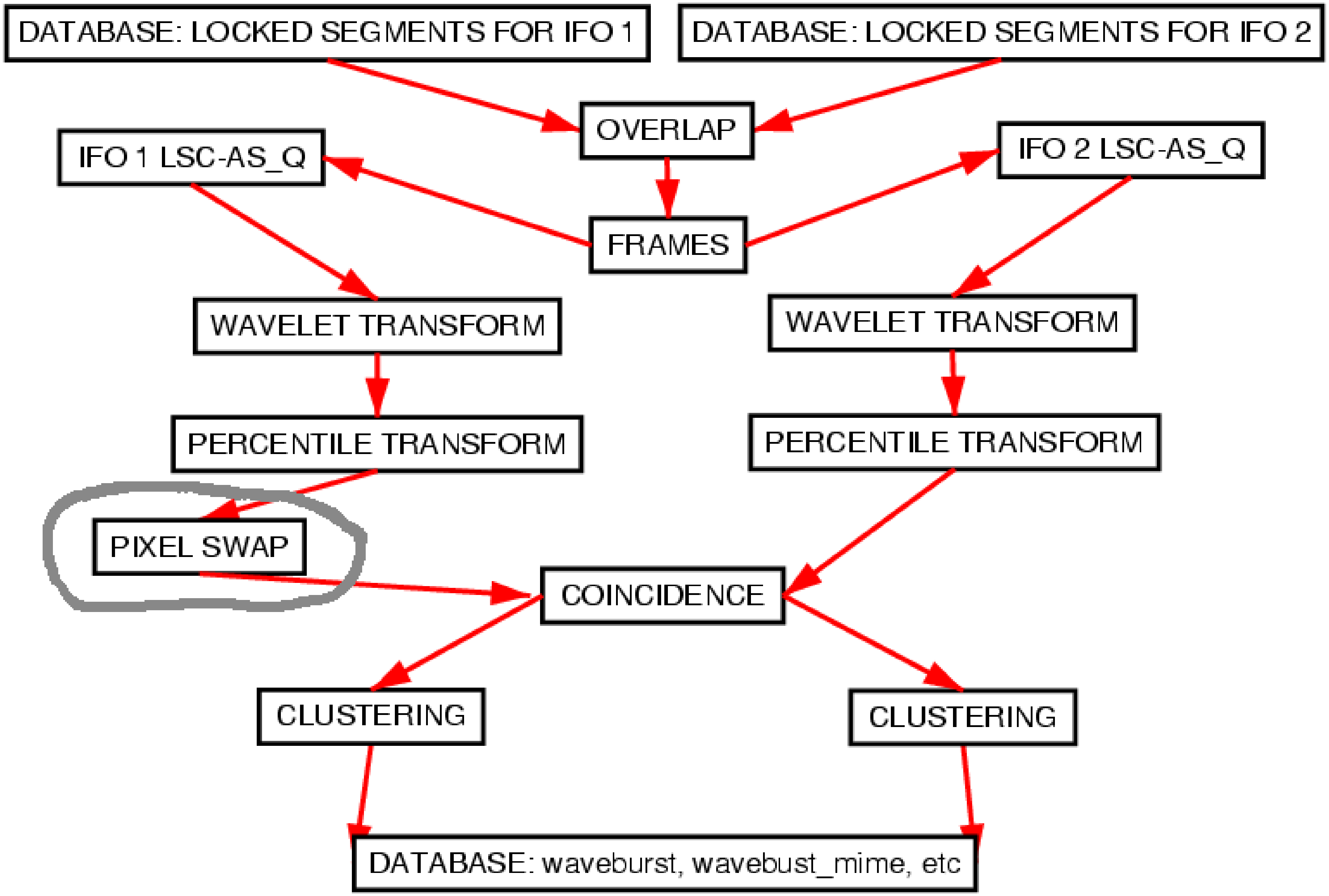

DATABASE: LOCKED SEGMENTS FOR IFO 1
DATABASE: LOCKED SEGMENTS FOR IFO 2
OVERLAP
IFO 1 LSC-AS_Q
IFO 2 LSC-AS_Q
FRAMES
datacon
WAVELET TRANSFORM
WAVELET TRANSFORM
PERCENTILE TRANSFORM
PERCENTILE TRANSFORM
COINCIDENCE
DSO
CLUSTERING
CLUSTERING
DATABASE: waveburst, wavebust_mime, etc

Data flow: event types

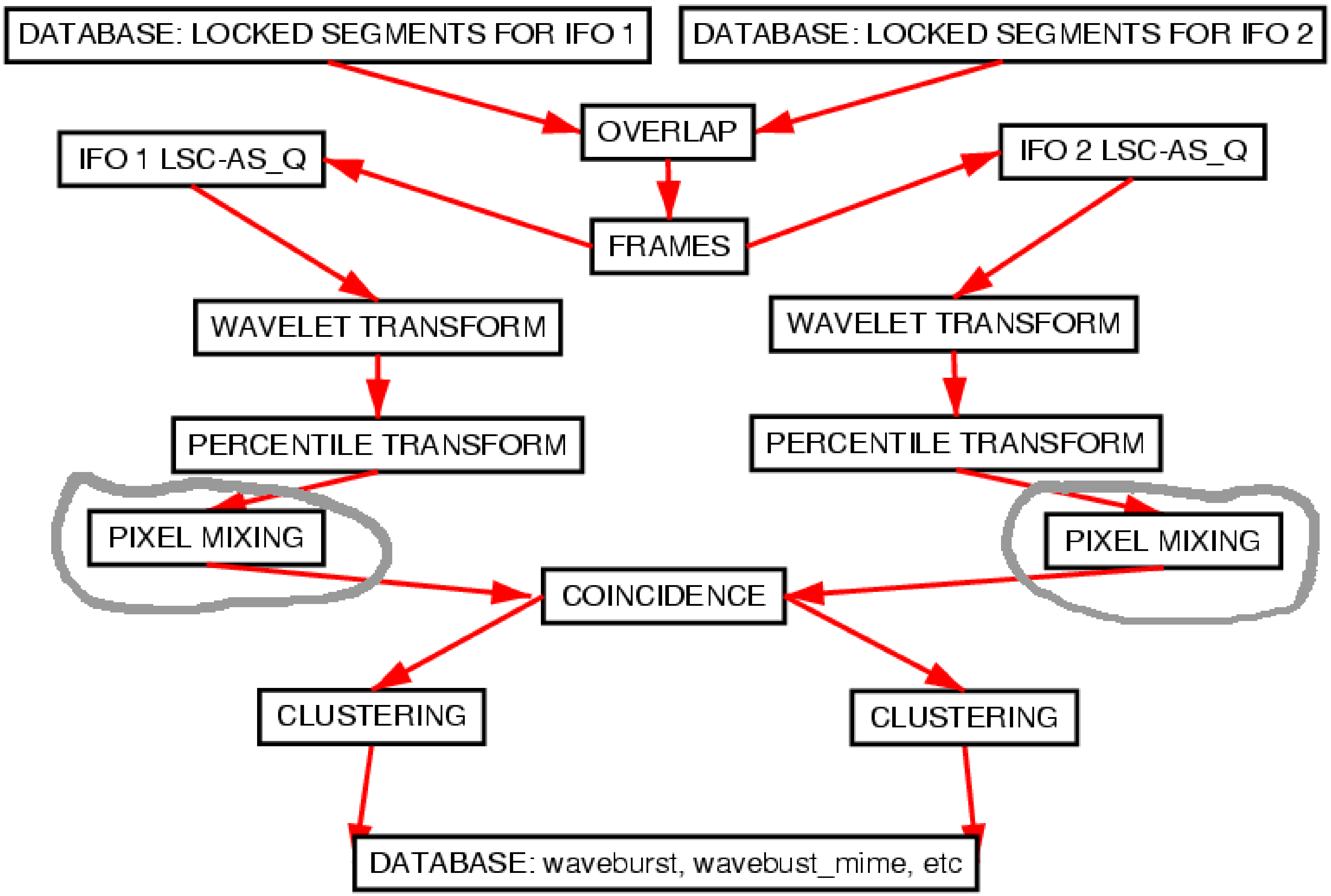


In a single run waveburst DSO is producing 3 kind of events:
gw candidate events,
offtime events (pixel swapping),
random events (pixel mixing).

Data flow: pixel swapping



Data flow: pixel mixing



Data flow

- **Percentile transform**: Select N% pixels with the biggest wavelet amplitude, zero out the rest and replace the original amplitude of the remaining pixels with a percentile amplitude.
- **Coincidence:** Select only pixels present in both channels within some time-frequency window, currently we use 0 time window (our time resolution is comparable to 10ms) and 0 frequency window. We are going to change that.
- **Clustering:** joining together pixels close in time and frequency and consider them to be a single event.
- **Pixel swap:** Estimating background event rate by exchanging in time two halves of data in each frequency layer.
- **Pixel mixing:** Estimating false alarm rate. Randomly change pixel positions in each channel for each frequency layer
- Waveburst can also work in a **single IFO mode** as all the other burst DSOs.

Input, performance

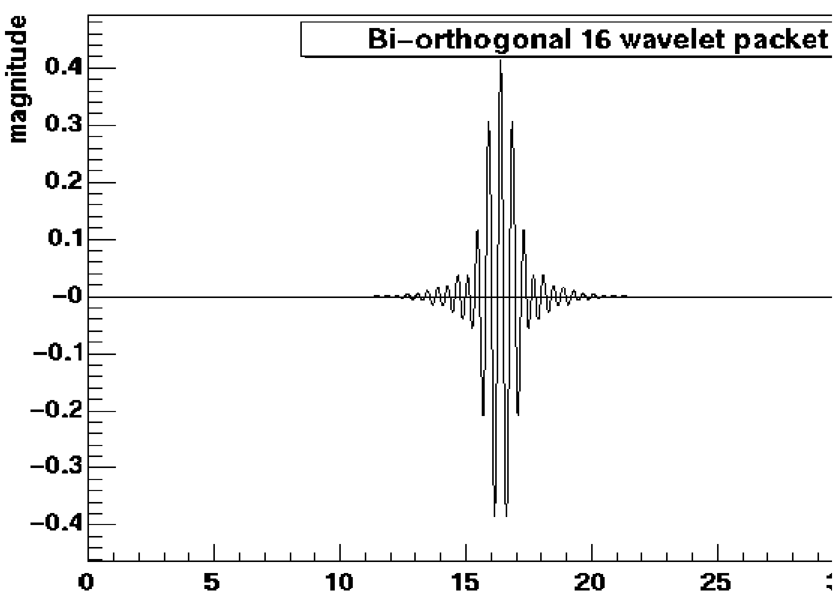
- LSC-AS_Q channel from a pair of interferometers.
- Currently use 120 seconds
- We used the same dewhitening filters for S1 as slope and tfclusters. No dewhitening filters are used for S2 so far.
- With the typical set of parameters waveburst runs about twice the real time for a pair of ifos on the existing hardware on one node.
- Typical total number of events found in 120 second interval is around 40.

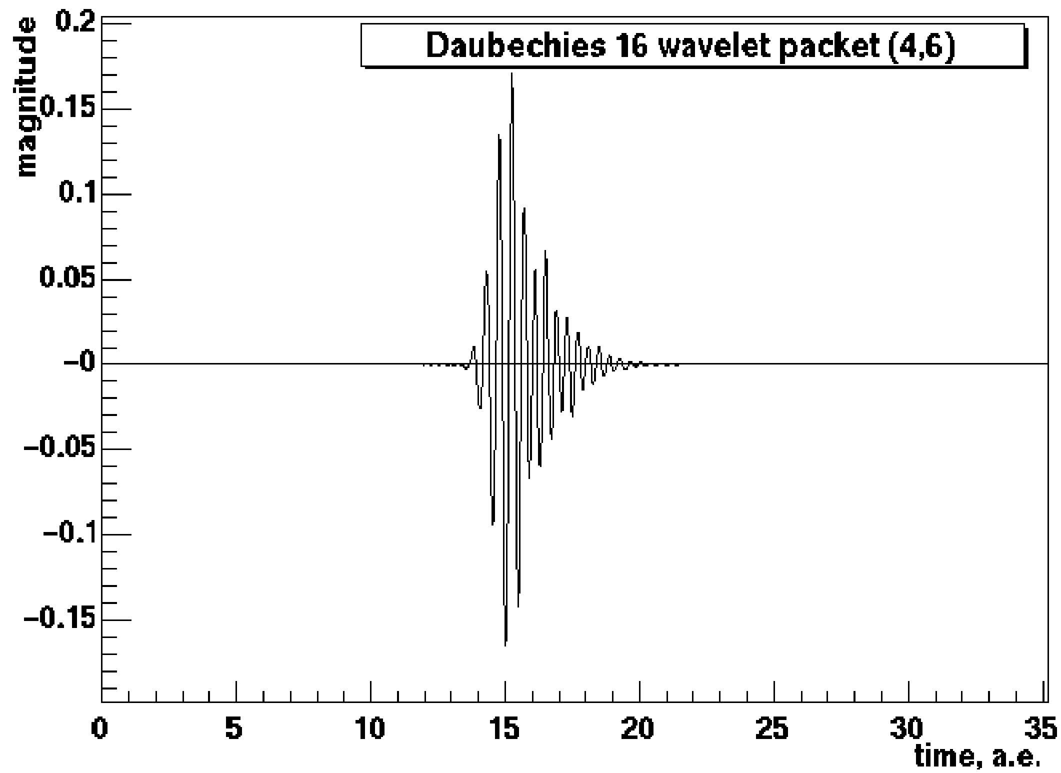
Filter parameters: main

name	value	comments
-ch1	L1:LSC-AS_Q	
-ch2	H1:LSC-AS_Q	
-nonZeroFraction	0.1	percentile fraction
-timeWindowNanoSec	0	coincidence time window
-freqWindow	0	to be implemented
-halo	TRUE	
-pixelMixer	TRUE	estimating false alarm rate
-seed	current time	to be used in pixel mixing
-pixelSwapOne	TRUE	estimating offtime rate
-pixelSwapTwo	FALSE	

Filter parameters: wavelet

name	value	comments
-WaveletType	BIORTHOGONAL	Can also use Daubechies, Simlet
-WaveletTreeType	BINARY	
-WaveletLevel	7	
-WaveletBorder	POLINOM	
-WaveletHPFilterLength	16	
-WaveletLPFilterLength	16	





Filter parameters: event selection

name	value
-minClusterSize	1
-maxClusterSize	+Infinity
-minLikelihood	10
-maxLikelihood	+Infinity
-minCorrelation	-1
-maxCorrelation	1
-minPower	0
-maxPower	+Infinity
-minEnergy	0
-maxEnergy	+Infinity

These parameters are used to reduce the output from DSO to the database. The most useful ones are minClusterSize and minLikelihood. They are chosen in advance independent of data depending on what kind of events we are looking for and what false alarm rate we can tolerate.

Filter parameters: blob output

name	value	comments
-blobP	TRUE	record a blob of percentile amplitudes for the smallest rectangle containing the cluster
-blobO	TRUE	record a blob of original amplitudes for the smallest rectangle containing the cluster

These parameters control the output of BLOBs to the database: for each event found we can record in the database the amplitudes (percentile or original) of the pixels in the smallest rectangle containing the cluster. This can be used, for example, to reconstruct the waveform of the signal later.

Filter parameters: debugging info

name	value	comments
-o1	FALSE	output original wavelet series for channel 1
-o2	FALSE	output original wavelet series for channel 2
-p1	FALSE	output wavelet series for channel 1 after percentile transform
-p2	FALSE	output wavelet series for channel 2 after percentile transform
-co1	FALSE	output wavelet series for channel 1 after applying coincidence
-co2	FALSE	output wavelet series for channel 2 after applying coincidence
-cl1	FALSE	output wavelet series for channel 1 after applying clustering
-cl2	FALSE	output wavelet series for channel 2 after applying clustering

Filter parameters: simulation

name	value	comments
-simulationType	0	not used at the moment
-simulationID		not used at the moment

The idea is to enumerate various simulation types and record their parameters to the database to be able to crossreference simulation type and parameters in waveburst (or any other DSO) tables.

Database tables: waveburst

1	CREATOR_DB	INTEGER	4
2	PROCESS_ID	CHARACTER	13
3	FILTER_ID	CHARACTER	13
4	EVENT_ID	CHARACTER	13
5	SIMULATION_ID	CHARACTER	13
6	SIM_TYPE	INTEGER	4
7	IFO_1	CHARACTER	2
8	IFO_2	CHARACTER	2
9	CHANNEL_1	VARCHAR	64
10	CHANNEL_2	VARCHAR	64
11	VOLUME	INTEGER	4
12	CORE_SIZE	INTEGER	4
13	XCORRELATION	REAL	4
14	RCORRELATION	REAL	4
15	LIKELIHOOD	REAL	4
16	POWER	REAL	4
17	MAX_AMPLITUDE	REAL	4
18	REL_START_TIME	DOUBLE	8
19	REL_STOP_TIME	DOUBLE	8

Database tables: waveburst

ROW	COLNAME	TYPENAME	LENGTH
20	DURATION	DOUBLE	8
21	START_TIME	INTEGER	4
22	START_TIME_NS	INTEGER	4
23	STOP_TIME	INTEGER	4
24	STOP_TIME_NS	INTEGER	4
25	START_FREQUENCY	REAL	4
26	STOP_FREQUENCY	REAL	4
27	BANDWIDTH	REAL	4
28	CLUSTER_TYPE	INTEGER	4
29	COMMENT	VARCHAR	64
30	S_CUT	INTEGER	4

Database tables

- **waveburst_mime** is used to record BLOBs of clusters amplitudes.
- **sim_type, sim_type_params, sim_inst, sim_inst_params** are intended to be used to enumerate the simulations so that it can be crossreferenced in the waveburst table. We are not doing it yet. For now simulations are done the same way as for slope and tfclusters. These tables are quite general purpose and can be used by any other DSO.

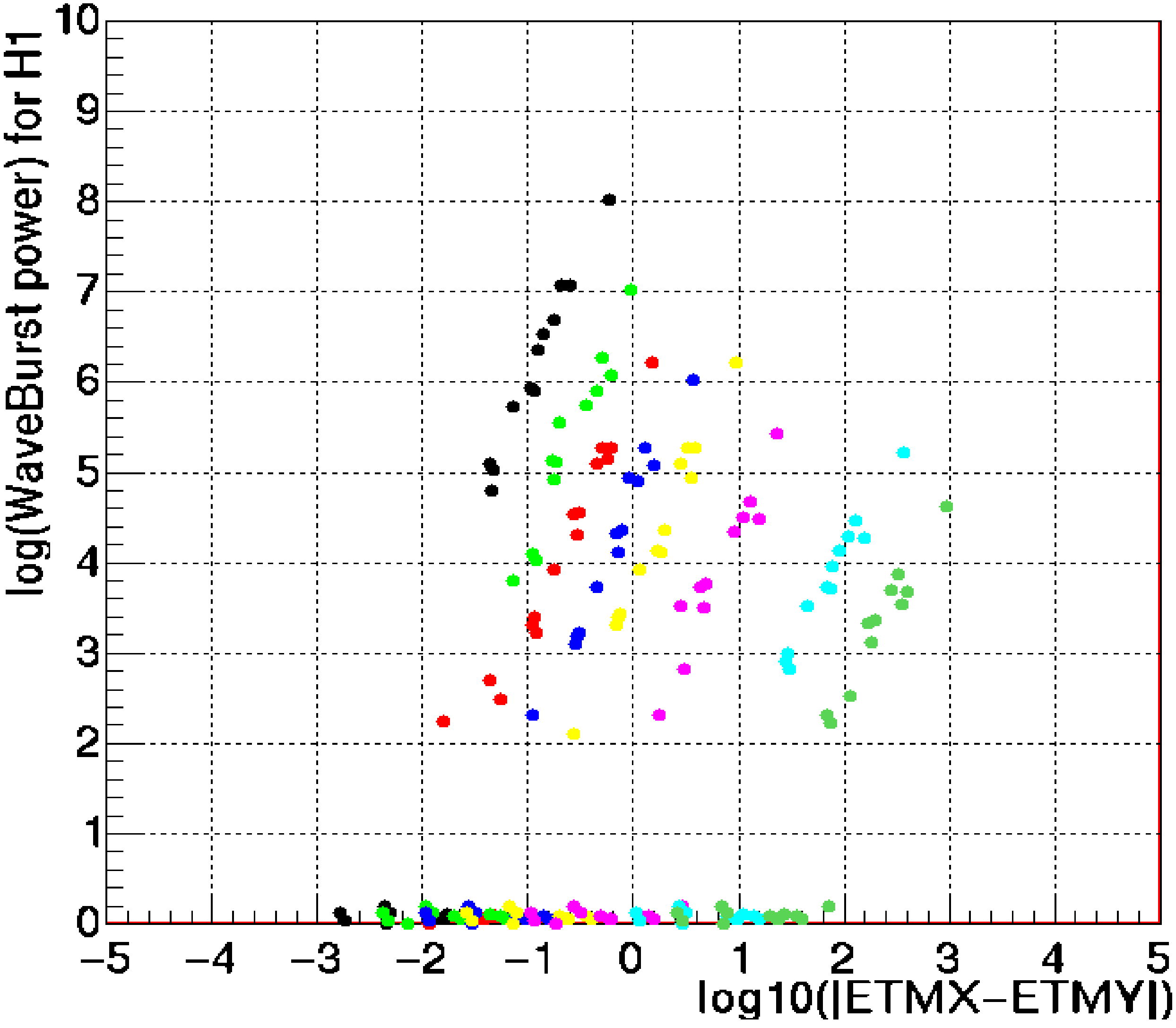
Detecting hardware injections. H1-H2 02/13/03

- Signal was injected to ETMX and ETMY simultaneously into H1 and H2.
- 8 frequencies: 100, 153, 235, 361, 554, 850, 1304, 2000 Hz.
- 24 strains for each frequency.
- Waveburst found on the time interval 729231185-729236099 334 events in each ifo. After the power cut (ln(power)>=2.5), 98 events survived in H1 and 95 events survived in H2.

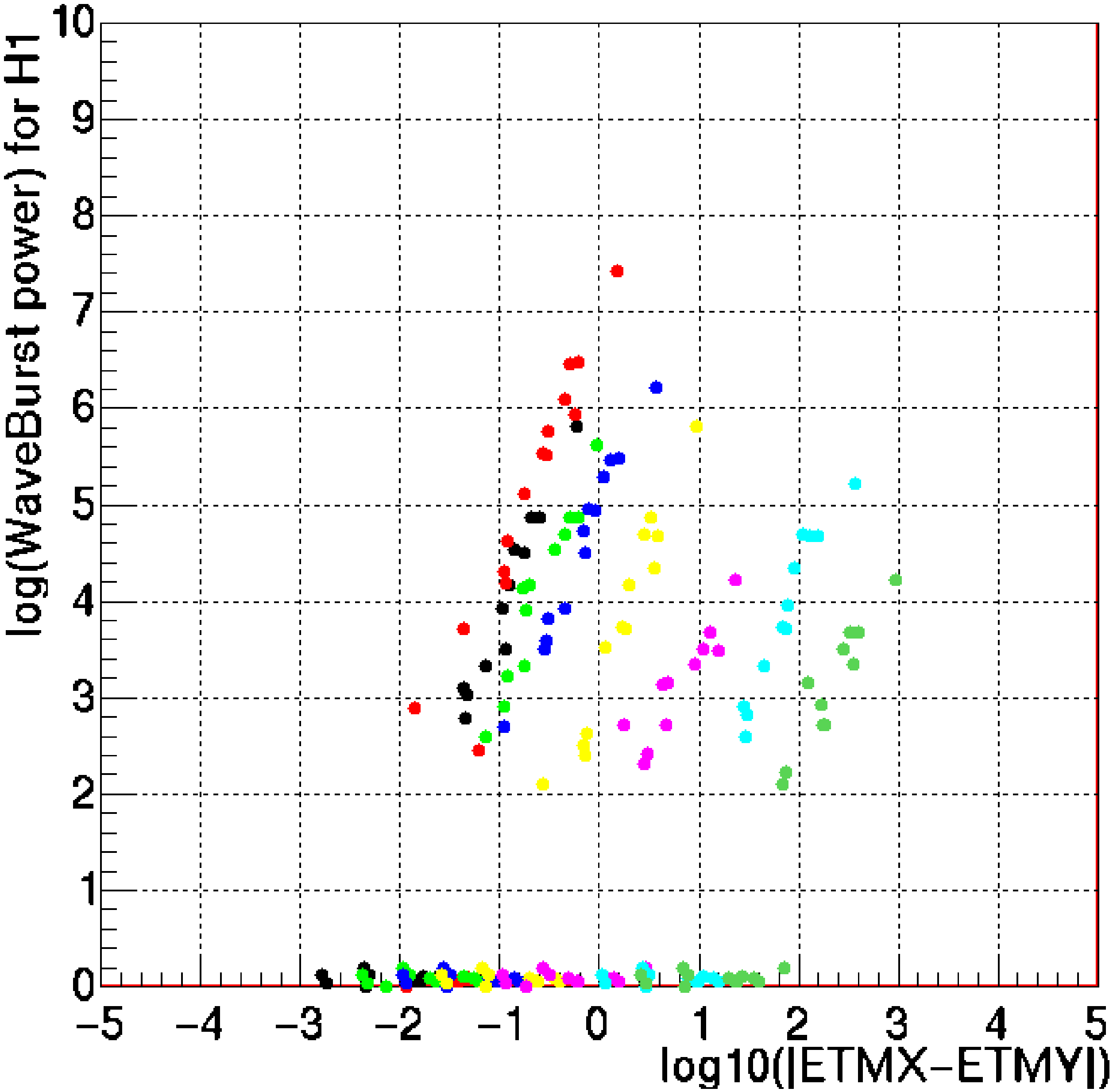
Detecting hardware injections. H1-H2 02/13/03

- No whitening filter was used. Whitening filter used by slope and tfcluster in S2 produced worse results. Might have something to do with different delays introduced by filters in different channels. To test, we need less strict coincidence.
- Approximately half of injections for each frequency was detected both in H1 and H2.

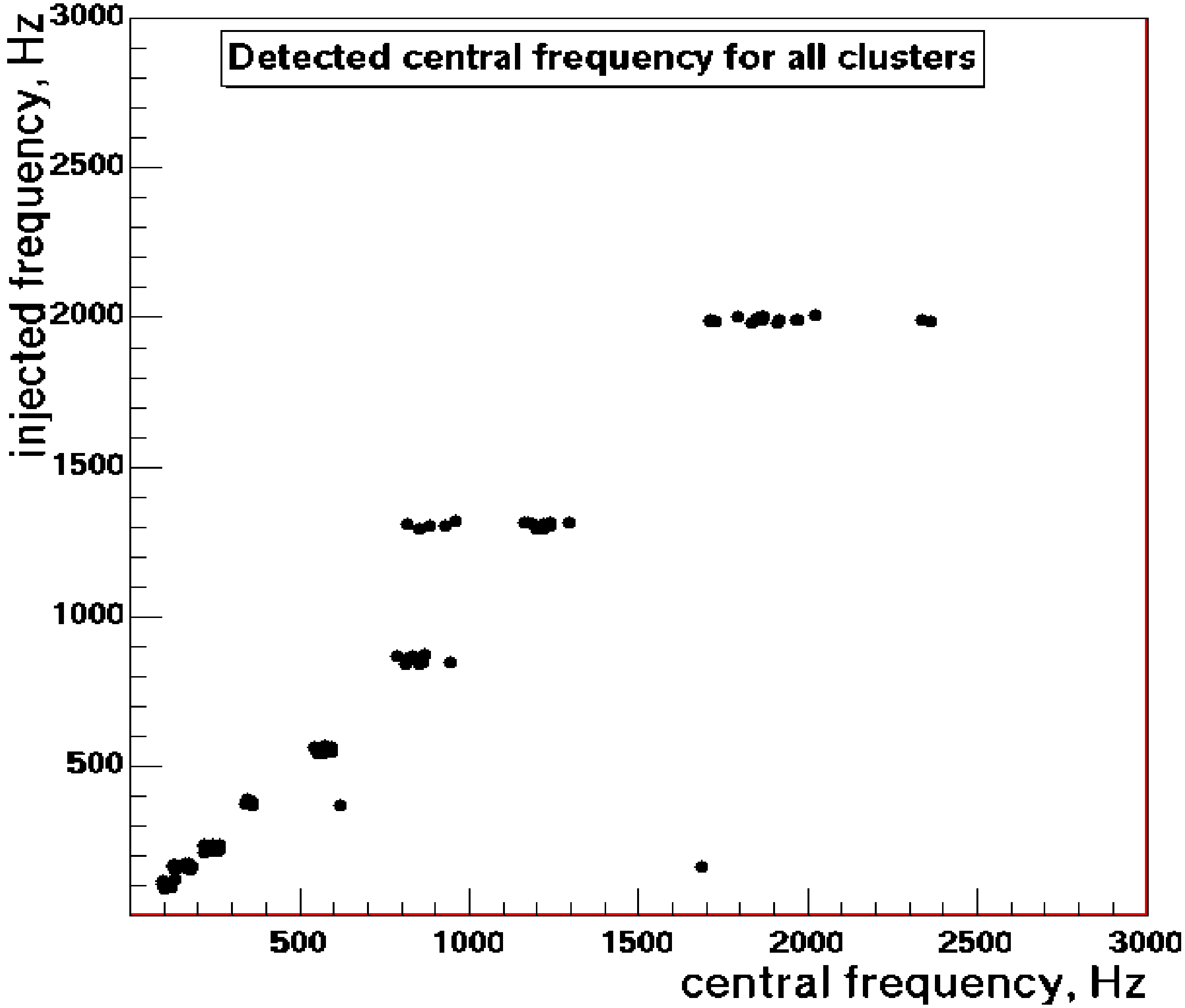
Detecting hardware injections: power in H1



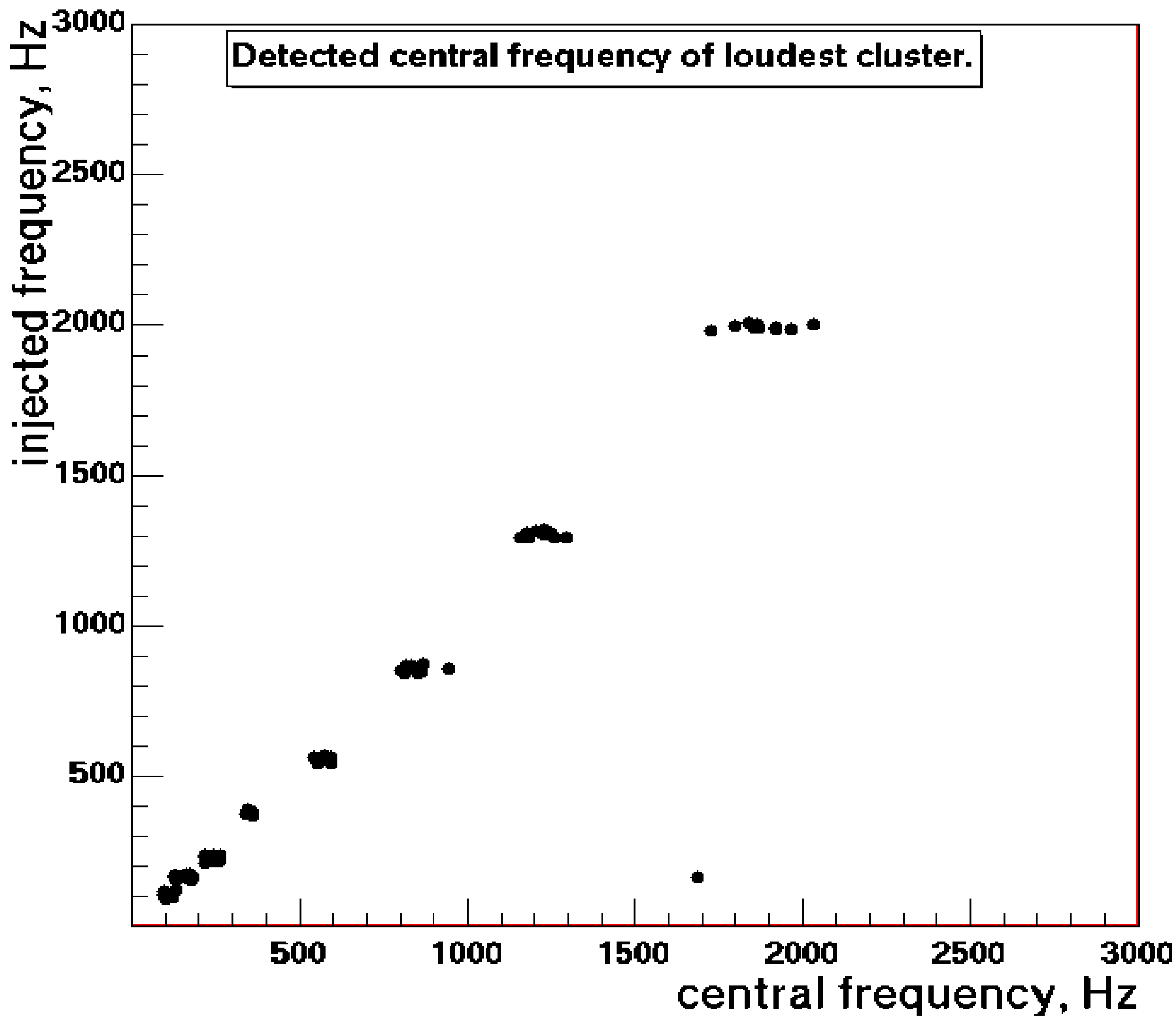
Detecting hardware injections: power in H2



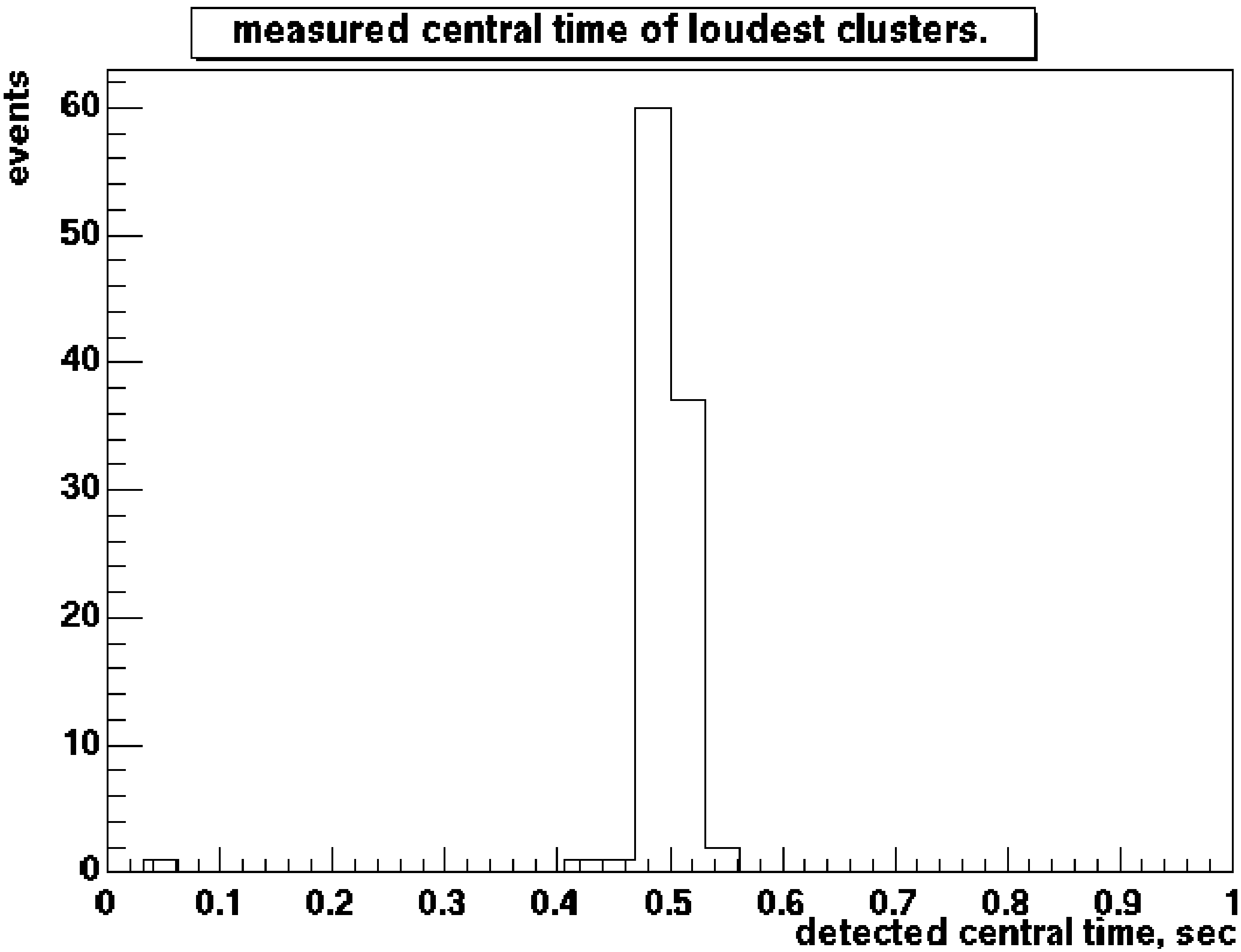
Detecting hardware injections: central frequency for all clusters



Detecting hardware injections: central frequency for loudest clusters



Detecting hardware injections: central time of loudest clusters



Future plans

- Loose coincidence using time and frequency windows.
- Process more than 2 ifos in a single job.
- Actually use the simulation tables in the database. Currently simulations are done the same way as is done for slope or tfclusters.
- After gw candidate events are found with the above procedure, apply correlation to further select only those events that have correlated counterparts in the other ifos.
- Analyze S2.

Future plans: correlation