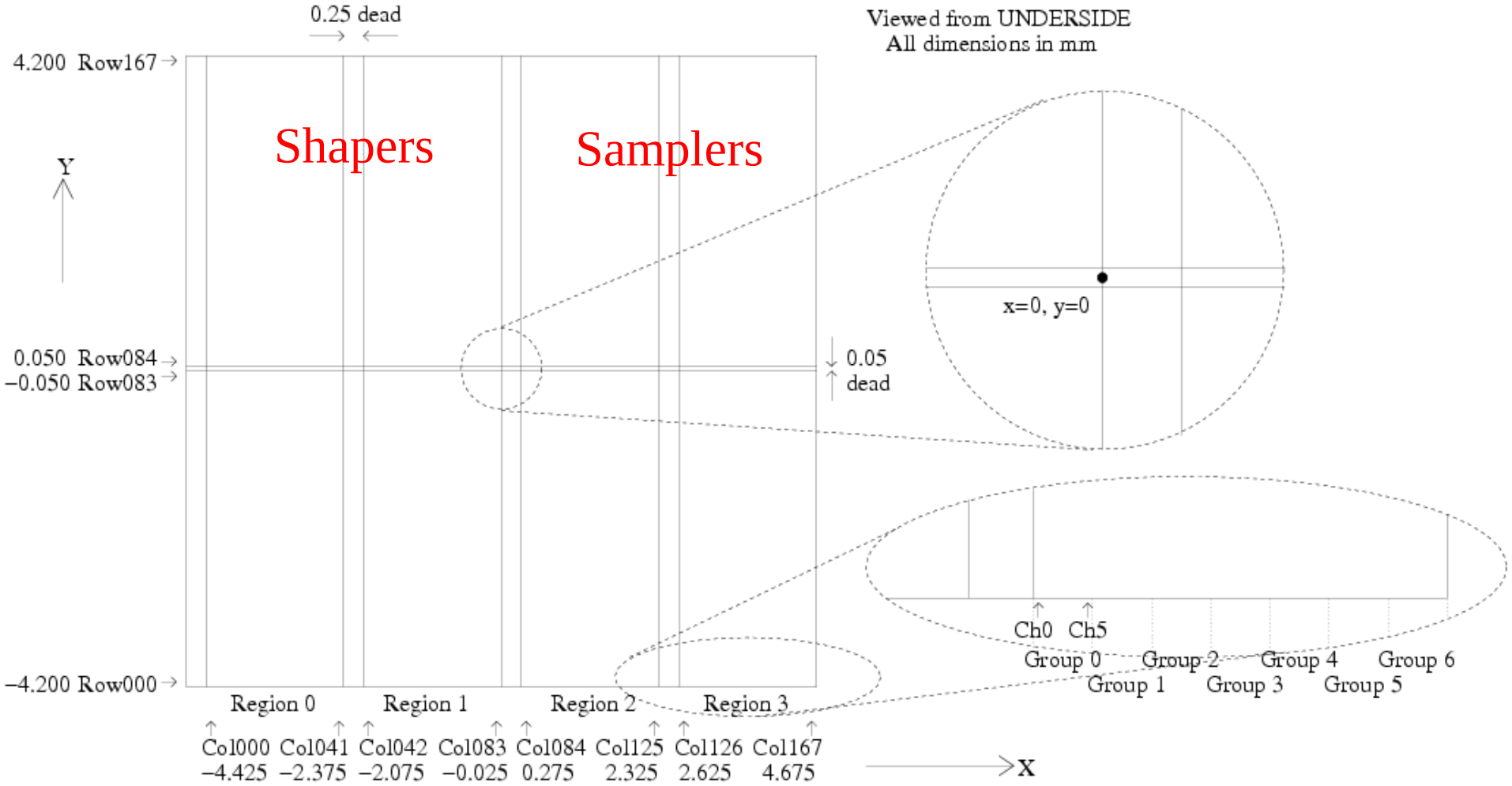
Test results from Imperial

- Basic tests
- Source tests
- Firmware status

Jamie Ballin, Paul Dauncey, Anne-Marie Magnan, Matt Noy

Imperial College London

A reminder of nomenclature

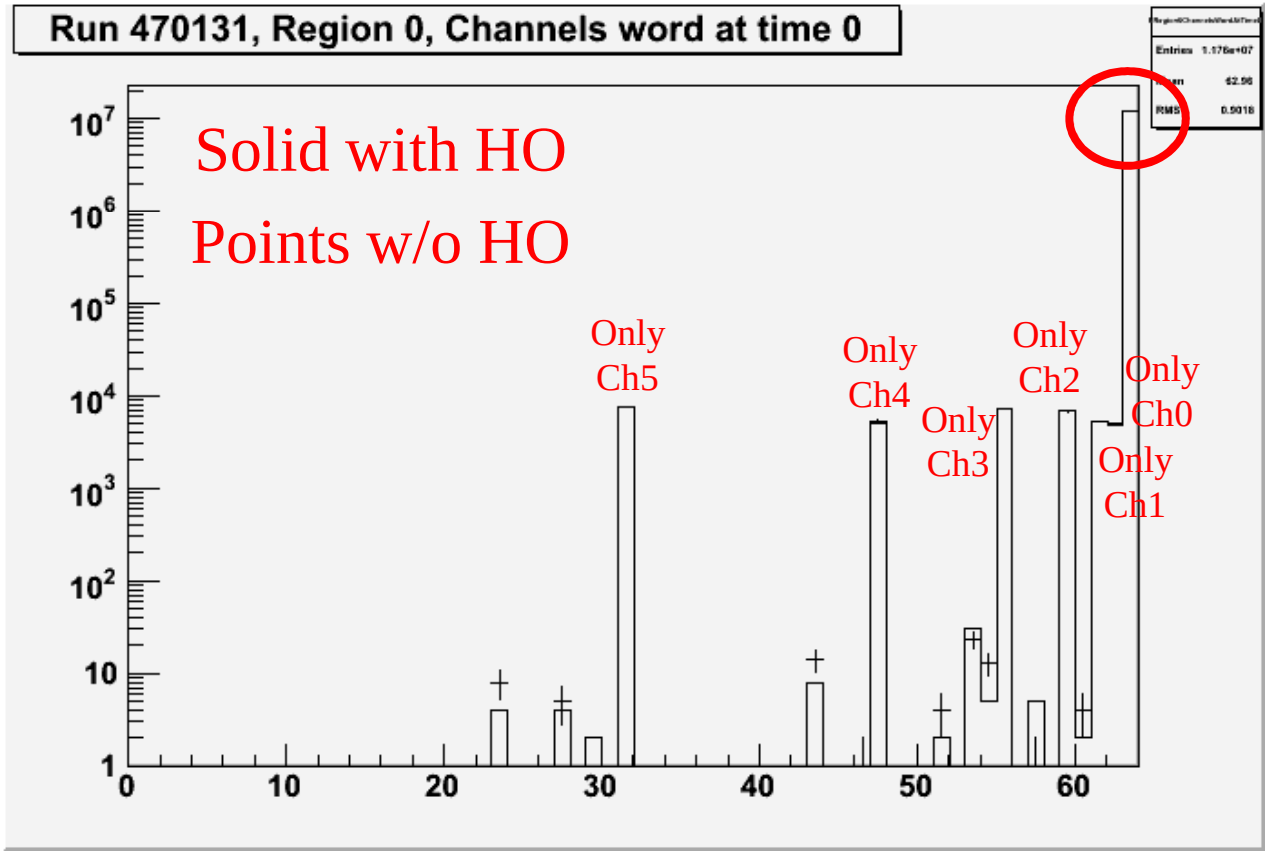


Major step forward: knowing a hit is a hit

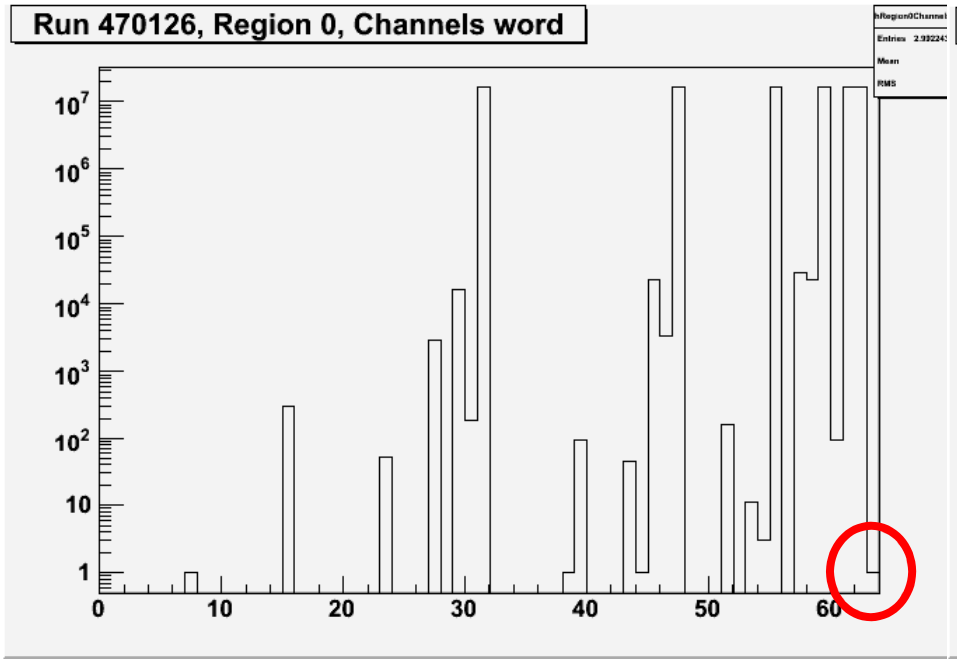
- A channel bit set in output word is NO hit
- HitOverride does NOT set all hits on
- 6 channel bits in word; values 0-63

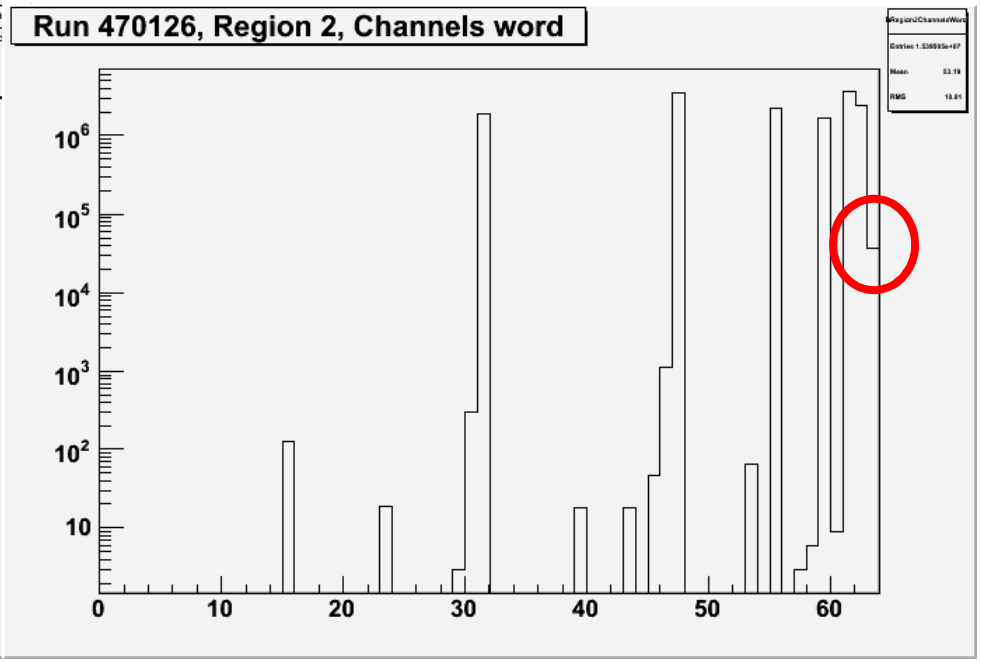
Only entries at 63 = no channels when HO set

All results throughout from sensor #2



Not always so clean...





These are without HO on so 63 should never appear

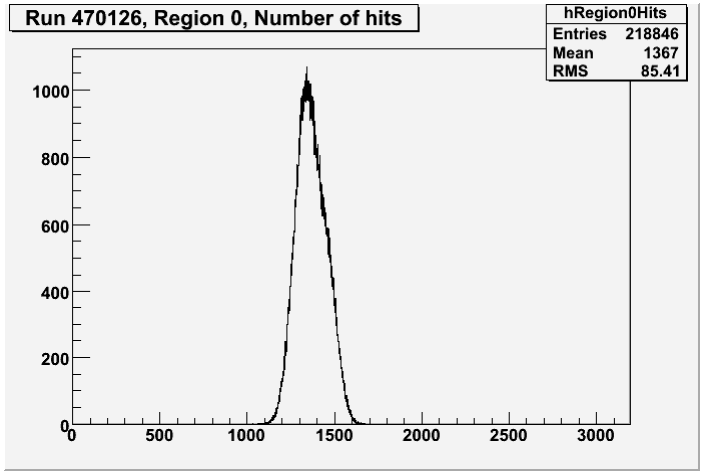
- Malfunctioning DataValid?
- Corrupt readout?

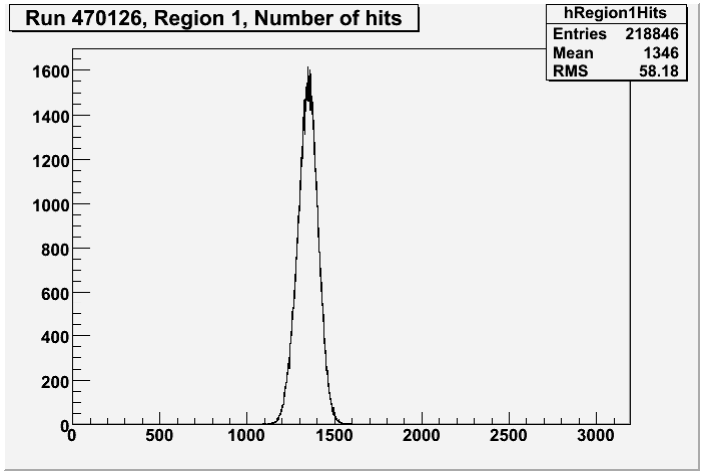
Could then diagnose “zero hit words”

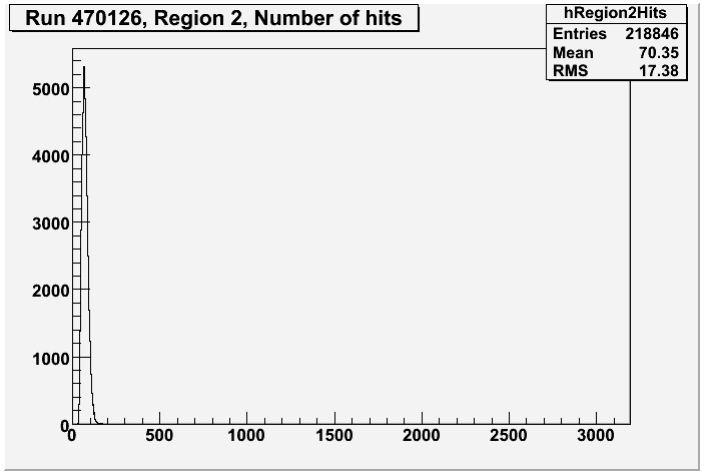
- Seen from regions 2 and 3
  - First 6 words from every row had no hits
  - Now understood as 6 words with ALL channels hit
- Due to injection of noise into pixels at start of bunch train
  - Adjust firmware timing to eliminate
  - Add MUX address 0 (“park” value) for HOLD switching
  - Now do not see these “zero” data
- Add delay from digital and analogue reset to start of bunch train
  - Gets rid of a lot of other noise...
  - ...but not all

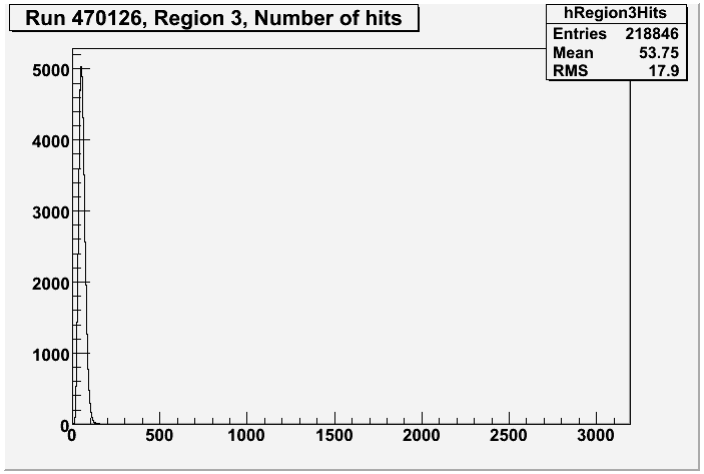
Noise studies

- Just read out pixels for bunch trains of 5000 beam crossings
  - Slower clock; each BX = 400ns
  - Thresholds: region 0 and 1 = 100, regions 2 and 3 = 50



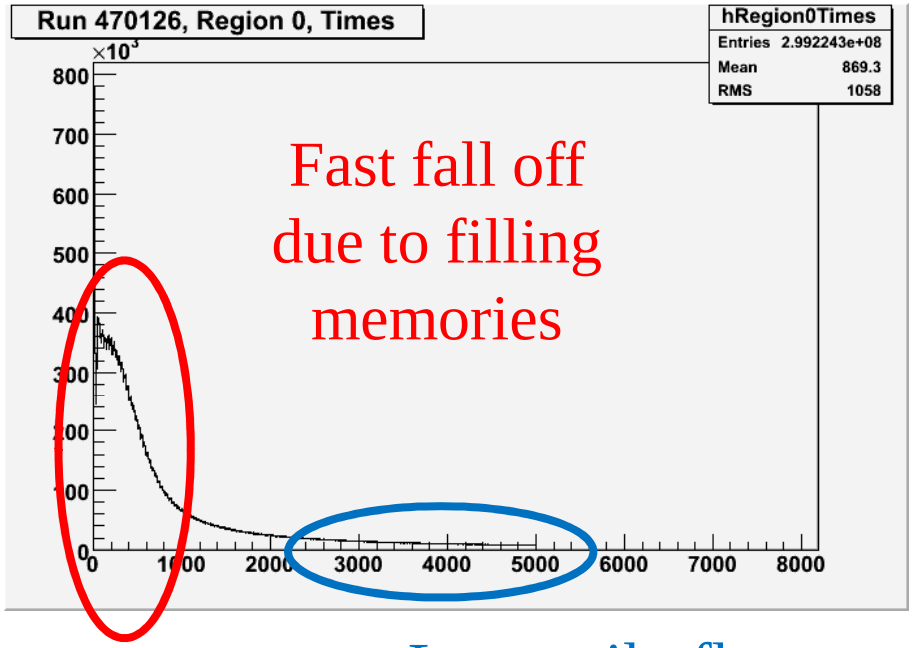


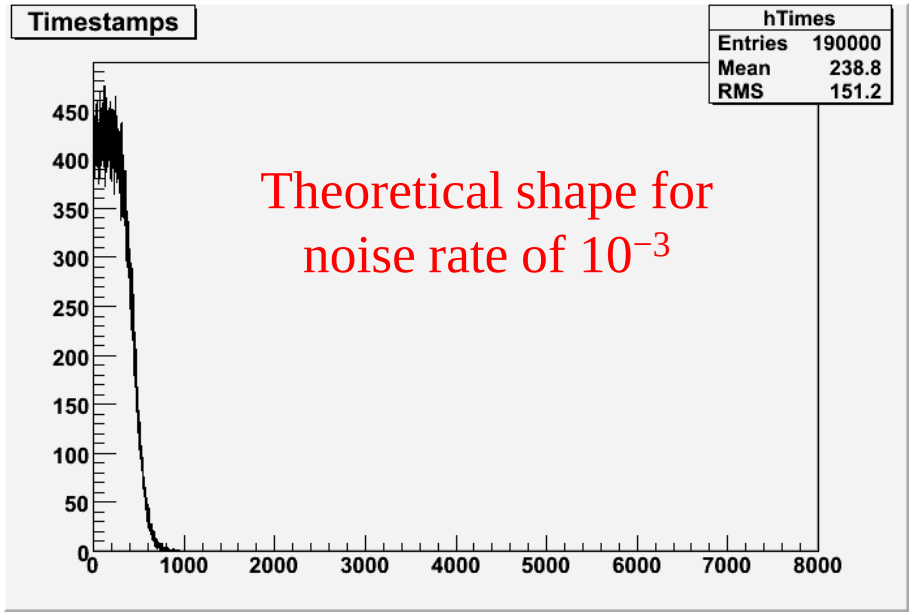




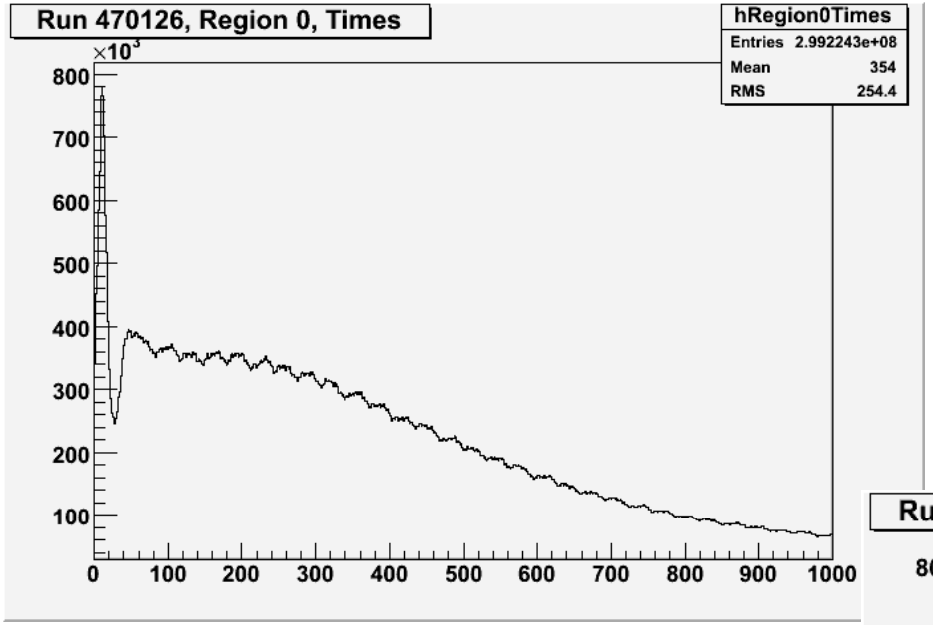
Time dependence for shapers

- Look at timestamps of all words



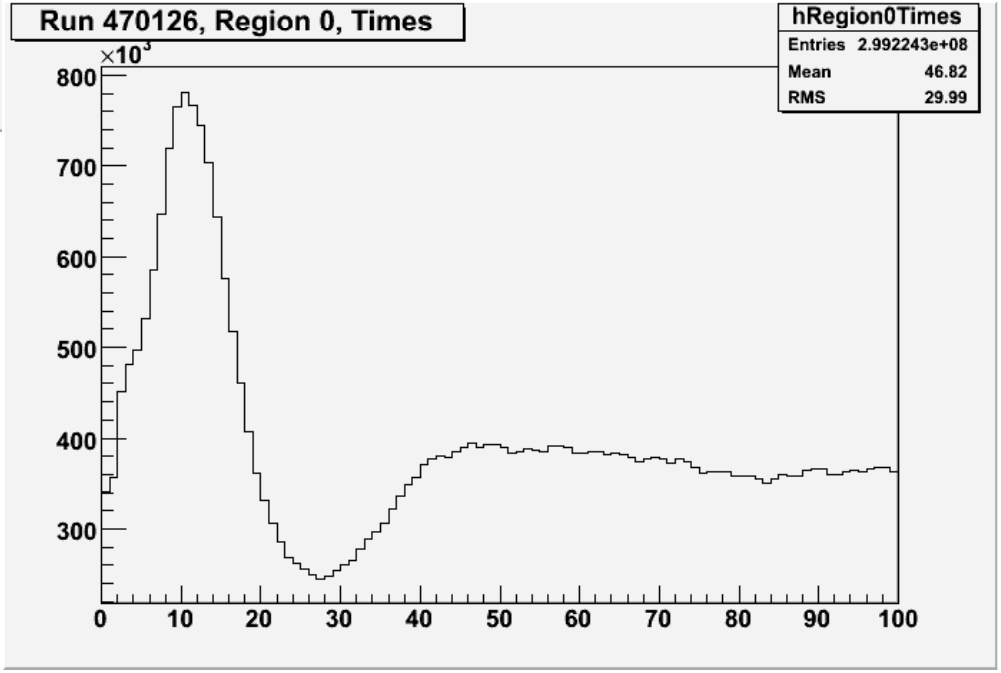


Details of early time dependence

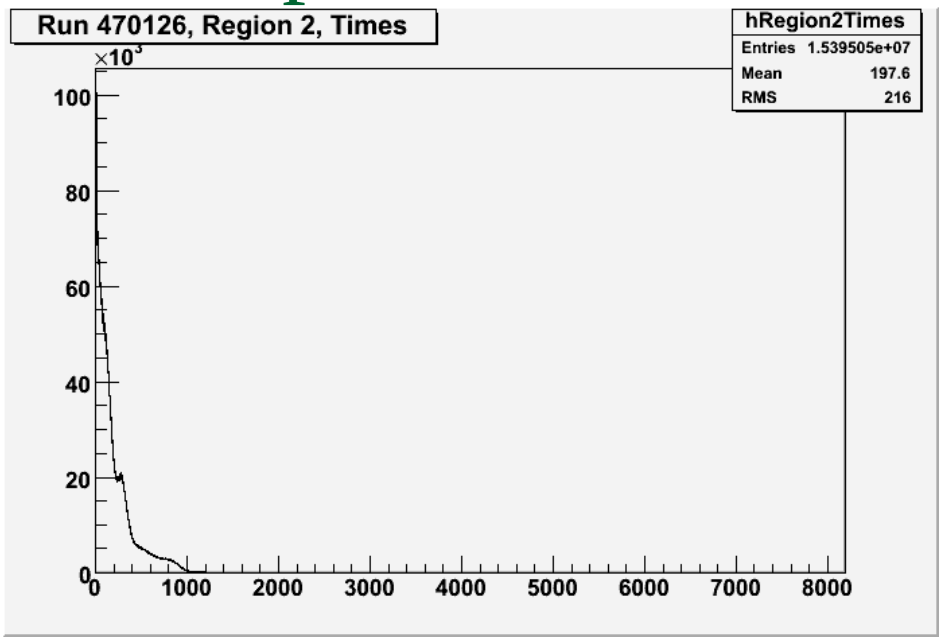


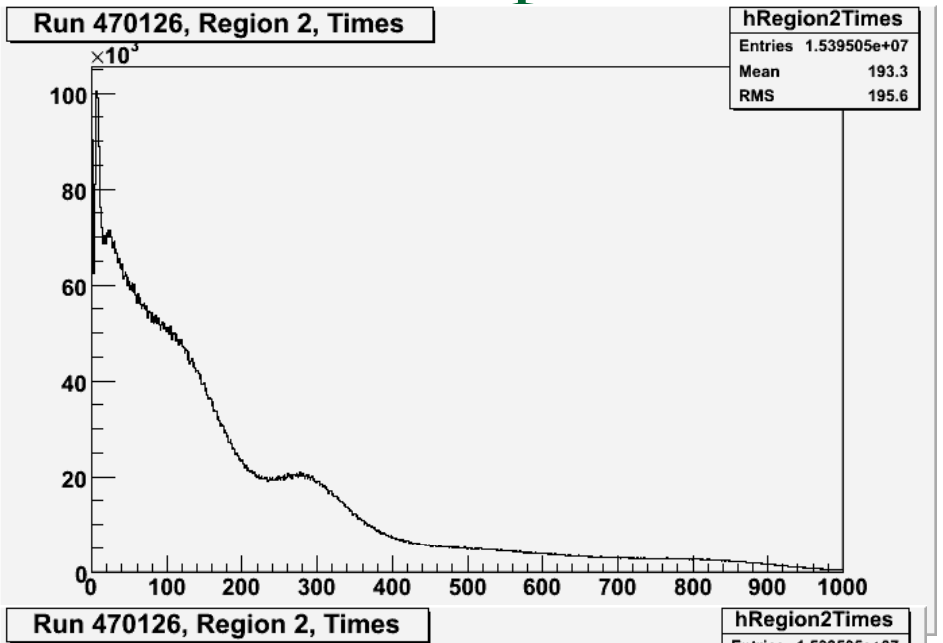
Early oscillation with period ~40BX = 16μs; presumed caused by clocks starting

Later oscillations with period ~32BX = 12μs; cause unknown

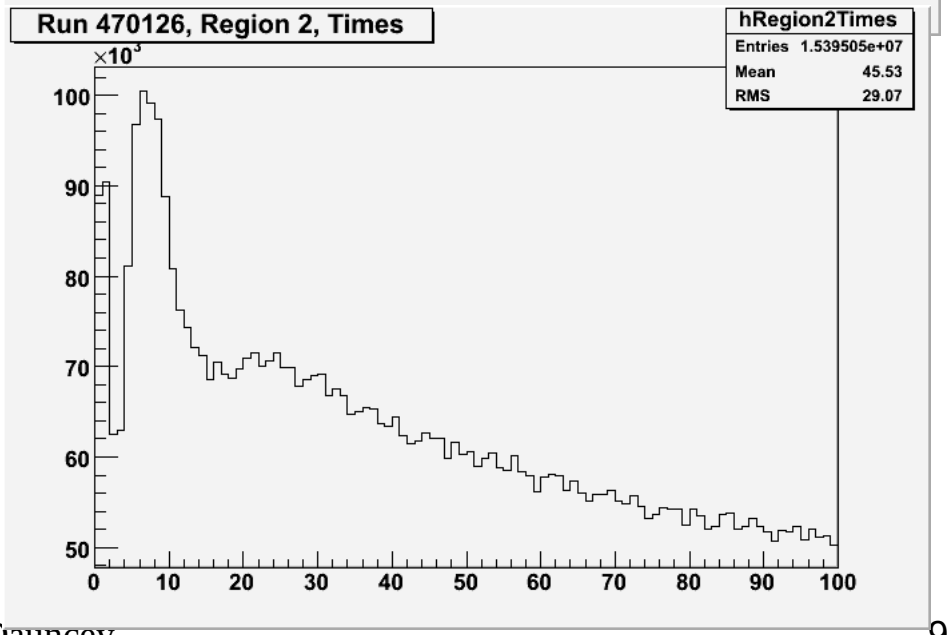


Samplers are more difficult to interpret

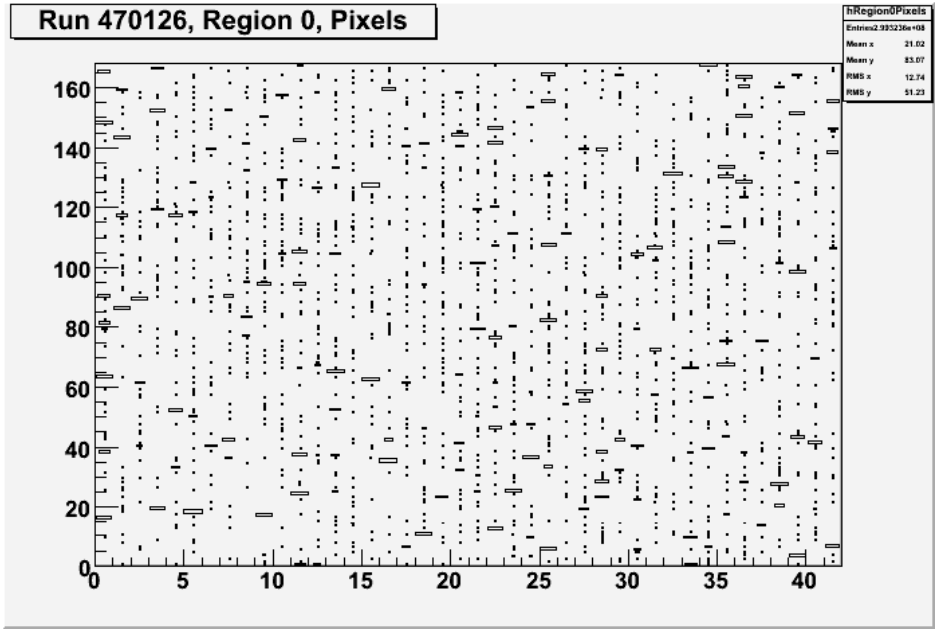


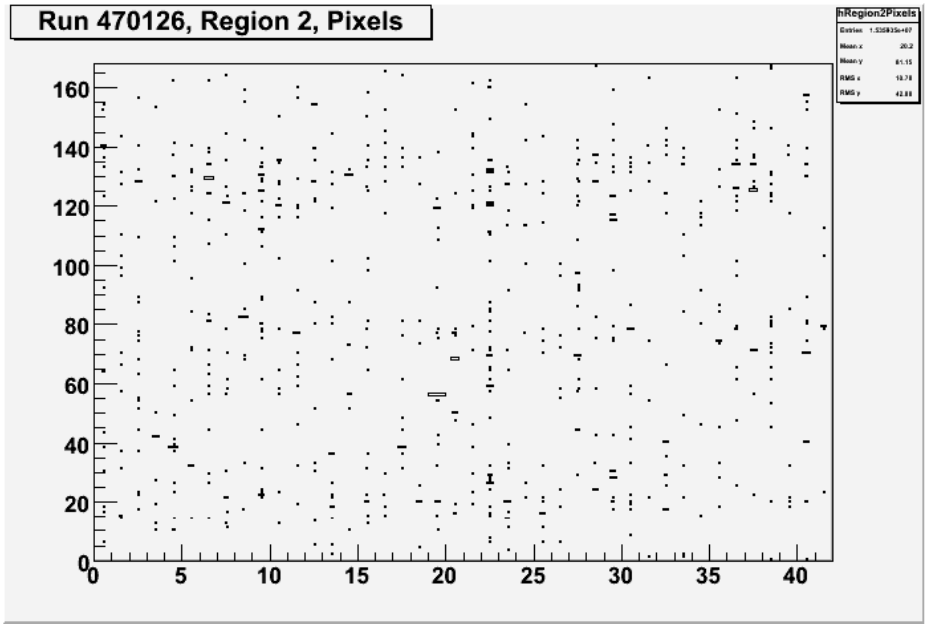


Faster falloff, even though number of words is smaller; saturation of sampler?



Locations of hits are not uniform

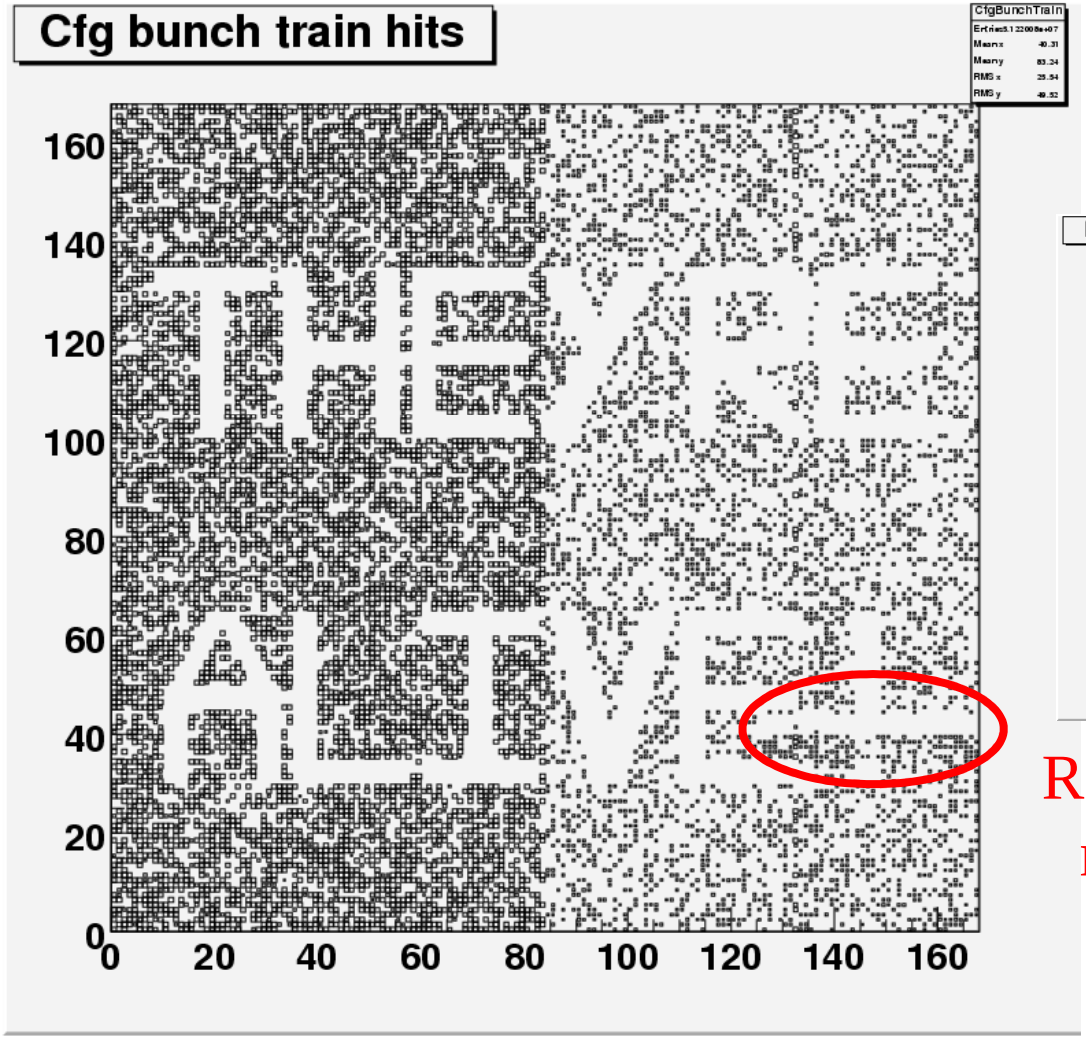


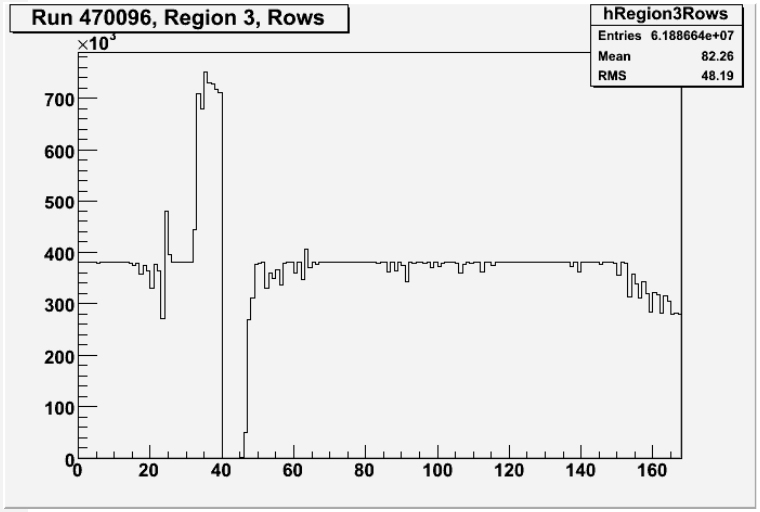


- ~500 channels (out of 28k) cause most of the hits

Should be able to mask these out

- On Friday got firmware to do masking working

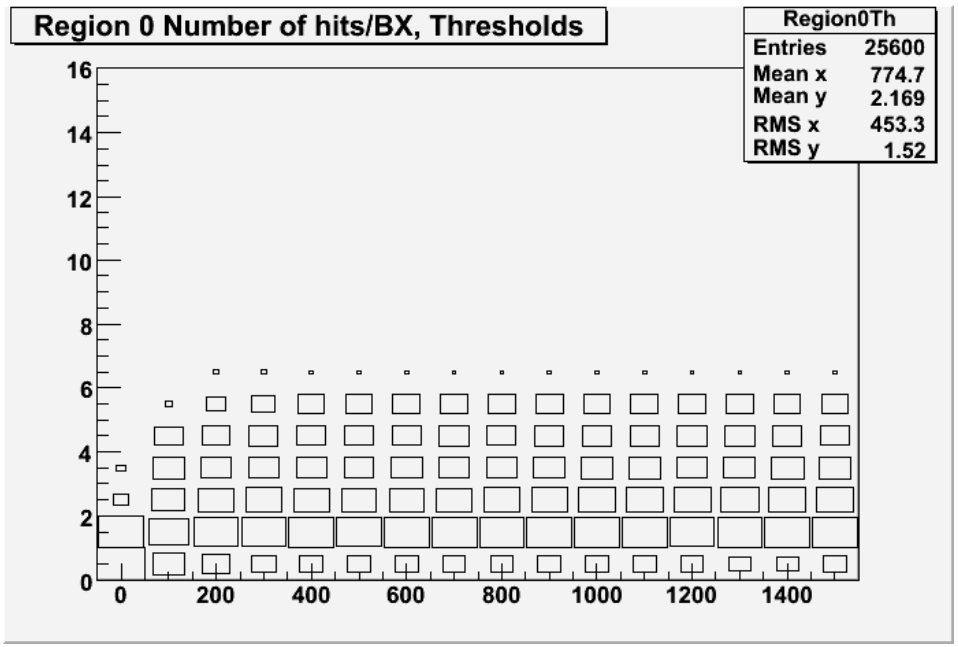


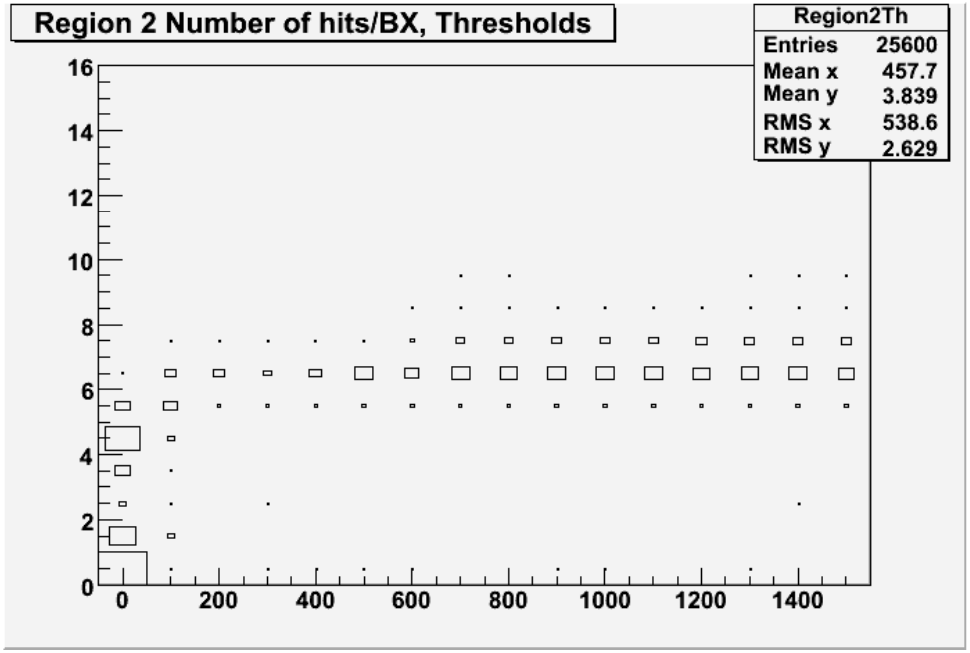


Readout error for one row bit in region 3

Can also now set trim thresholds

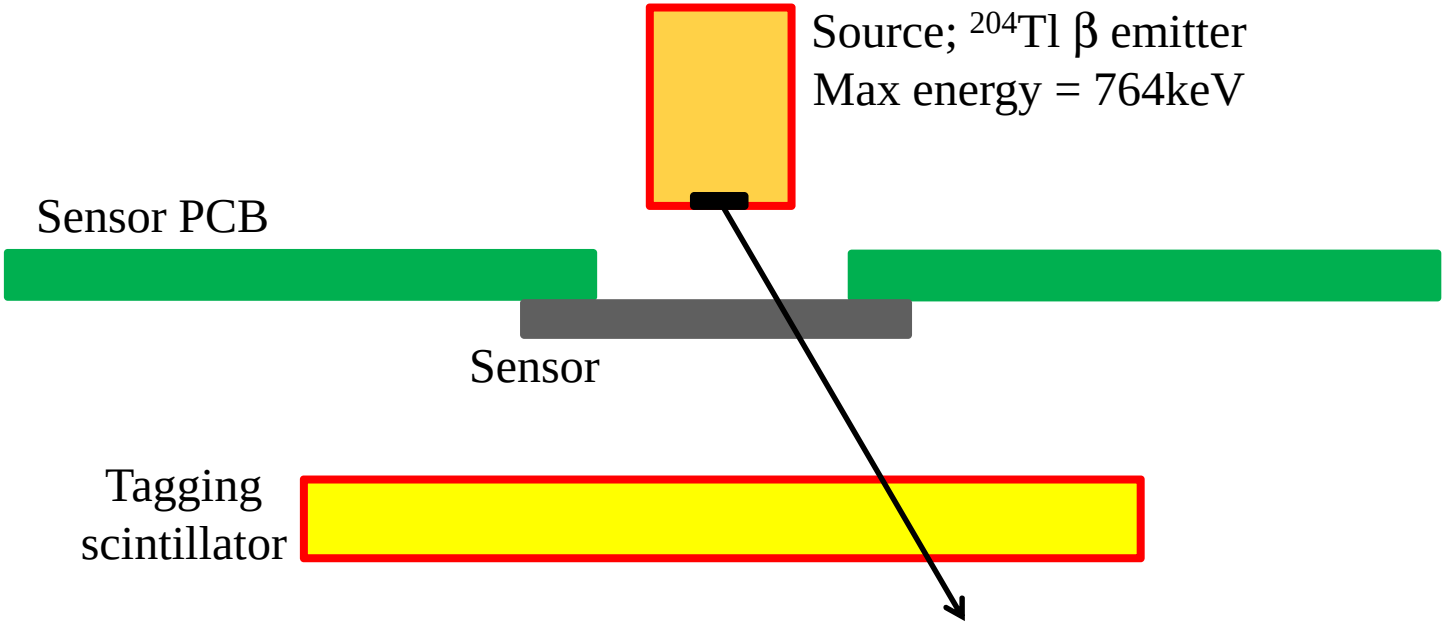
- Scan both overall and trim threshold settings
- Simply plot average number of hits
  - Look for line of constant hit rate to cross-calibrate





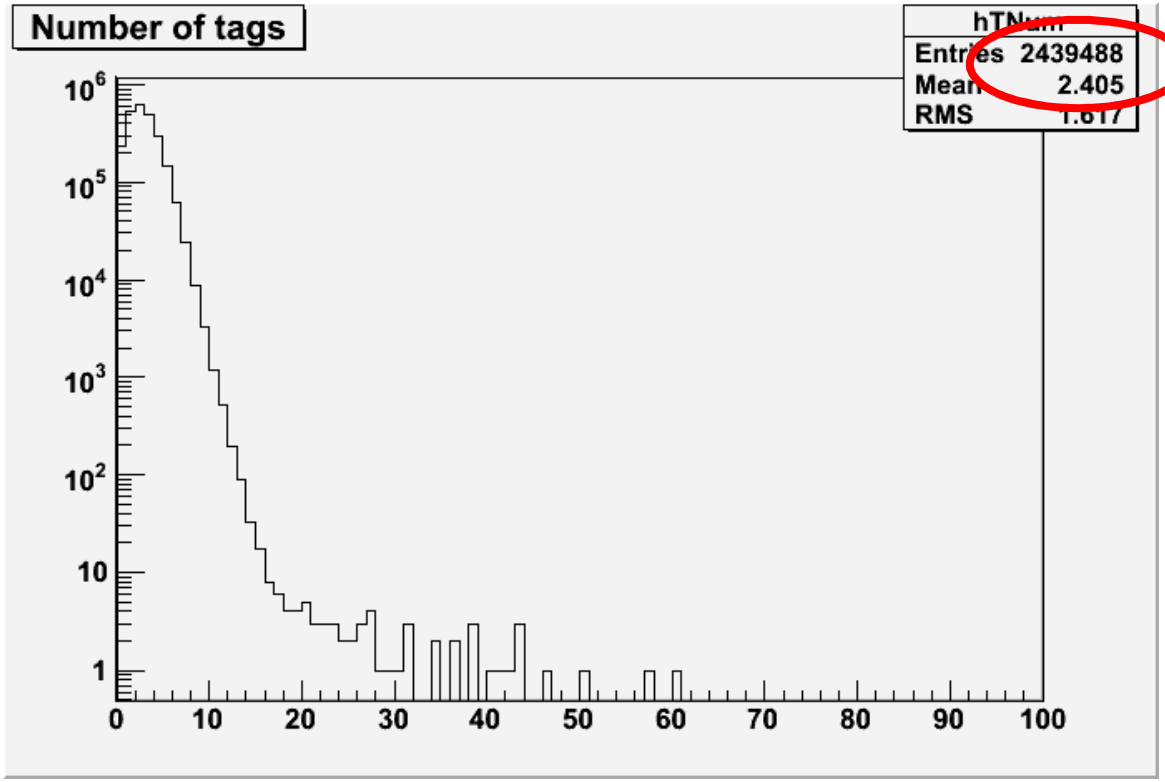
Do not understand these distributions at all...

Source tests



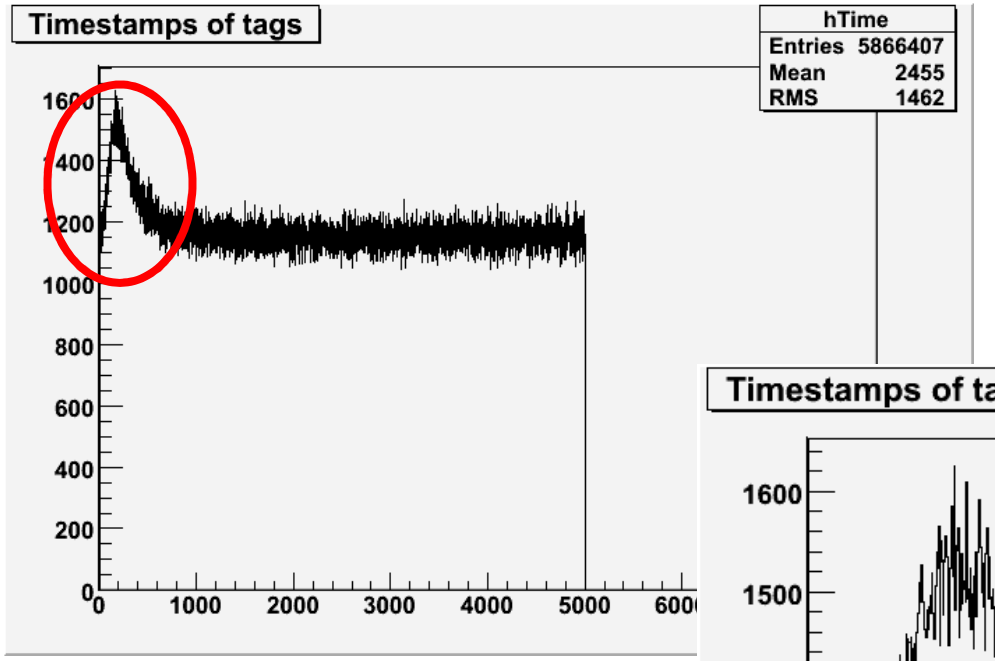
- Tagging scintillator read out via PMT
  - Out level latched and recorded every BX
- Signal is hit in sensor in coincidence with scintillator hit
  - Took runs so far without masking working so all noise hits

PMT time tags

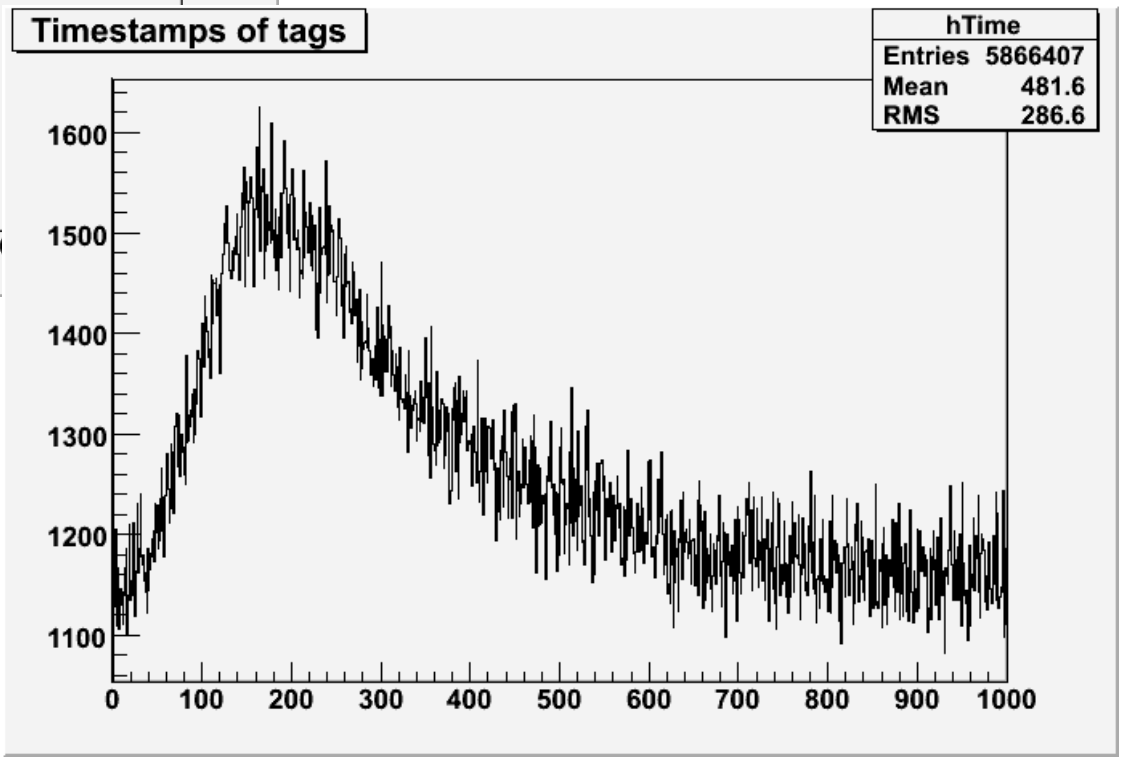


- Total number of bunch trains ~2.5M
- 12 hours of data-taking, average 55Hz
- Average rate of 2.4/bunch train ~1.2kHz
- Total number of tags ~5.9M

PMT tag time distribution

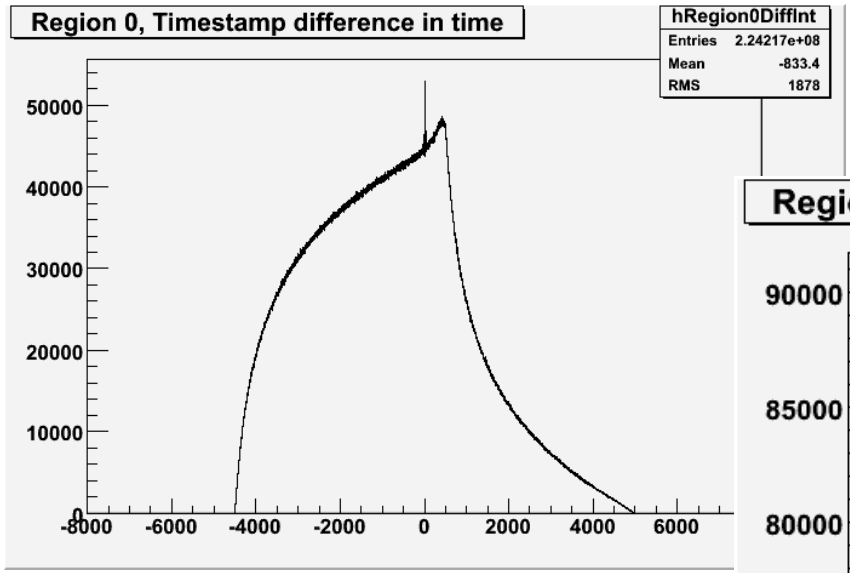


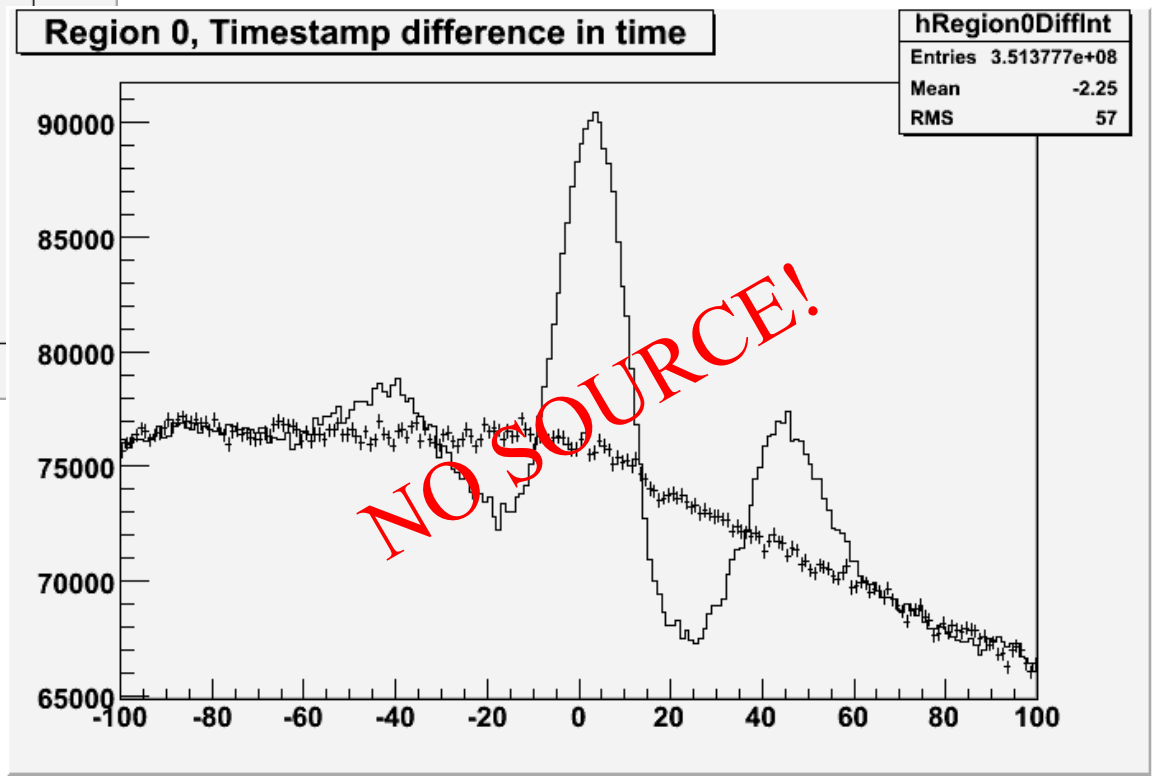
Clearly unphysical; related to start of bunch train so induced by sensor or DAQ



Compare tag and sensor hit times

- Take difference of timestamps

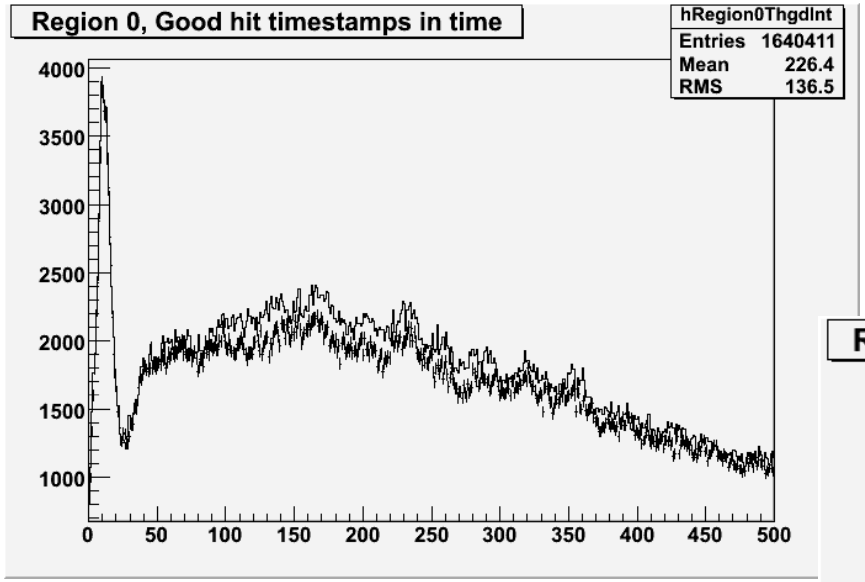


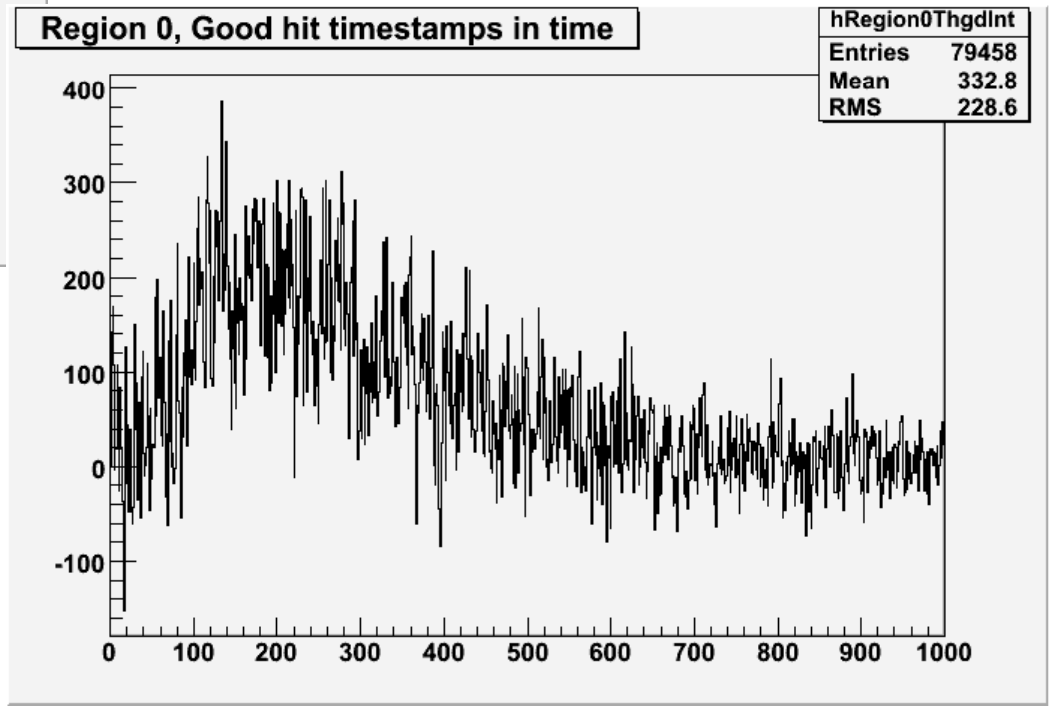


- Background estimate from same analysis but using tags from previous bunch train

Where are spurious hits coming from?

- Select peak and compare to background

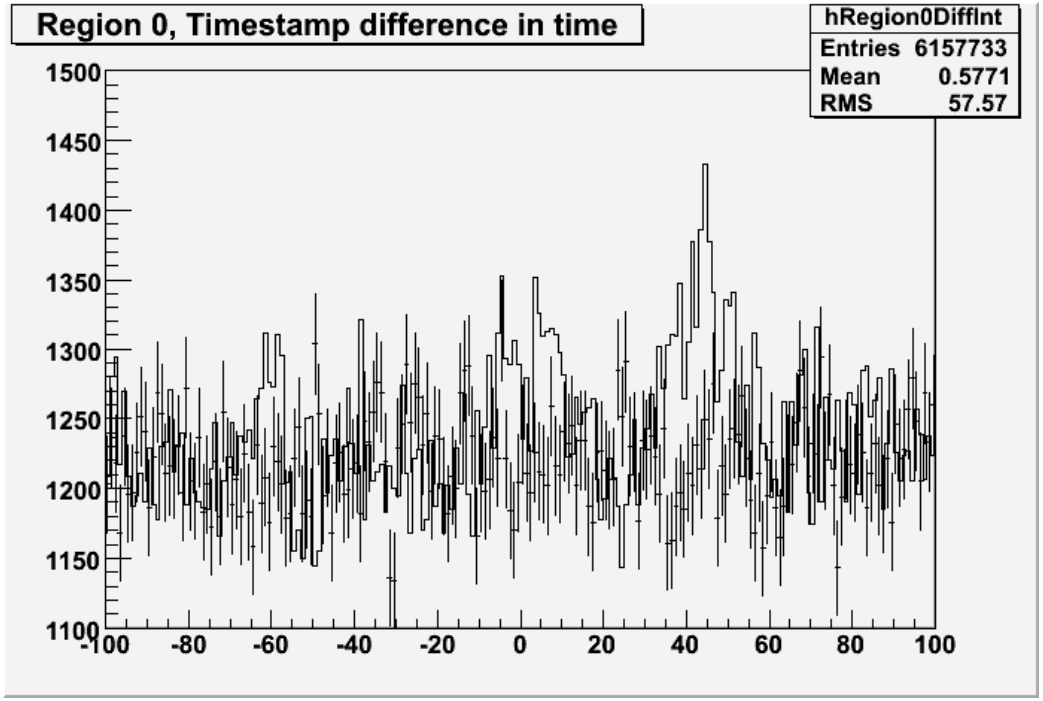




- Subtract off background
- Same shape as unphysical PMT hits; induced by a signal at bunch train start

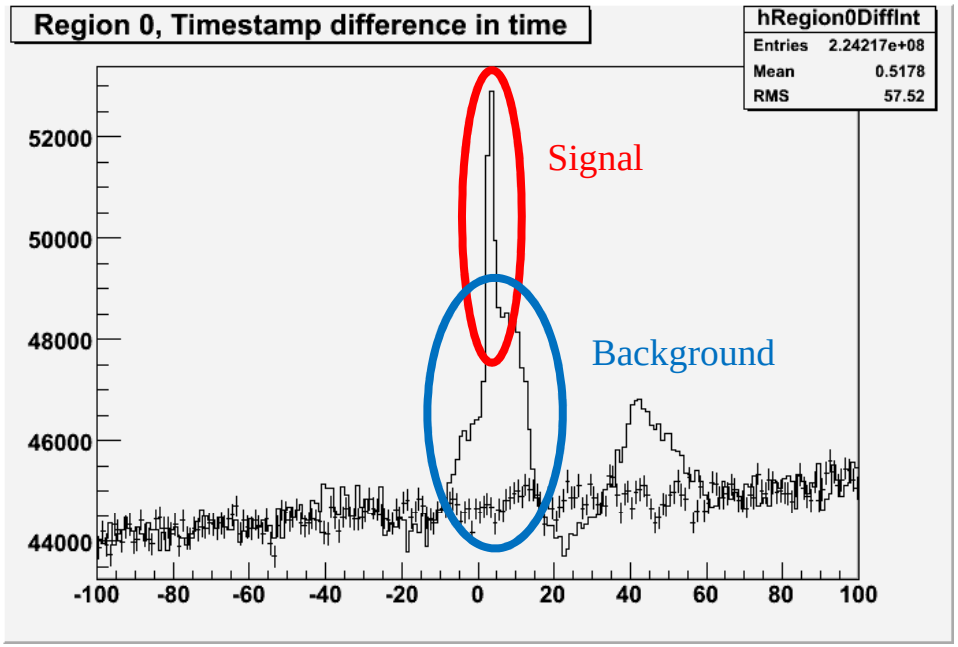
Cut out bump in time

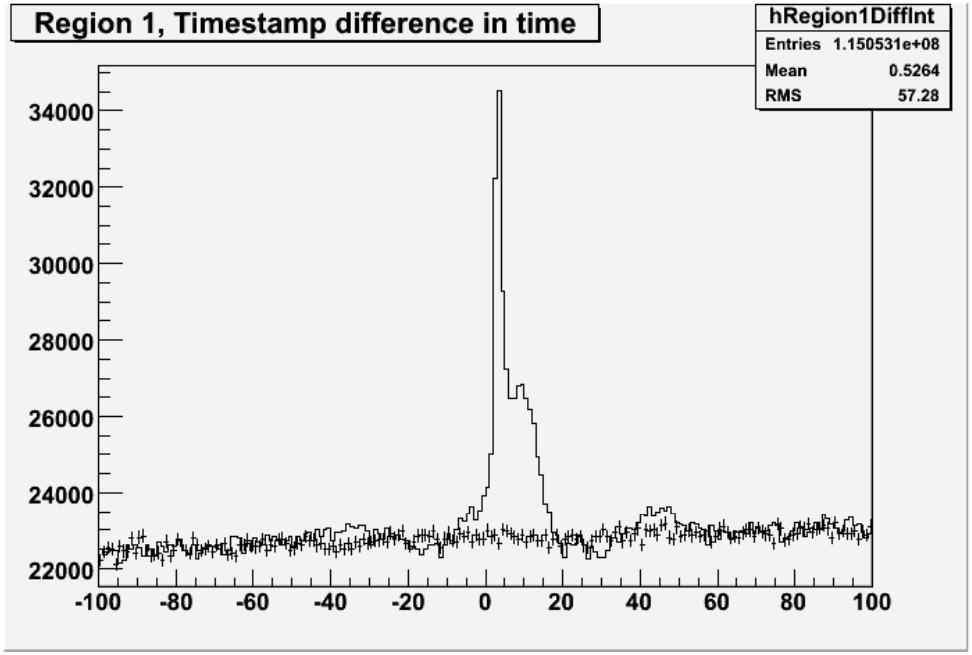
- Remove all timestamps < 500
- Also remove the ~500 bad channels by hand



- "No source" case looks better
- Still some residual peaks

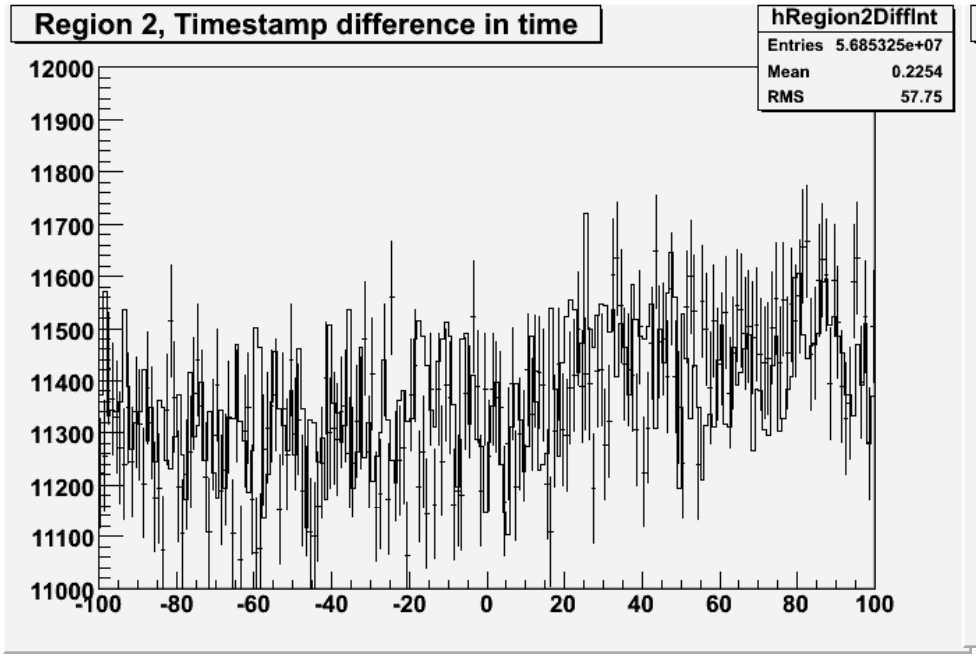
Now with the source

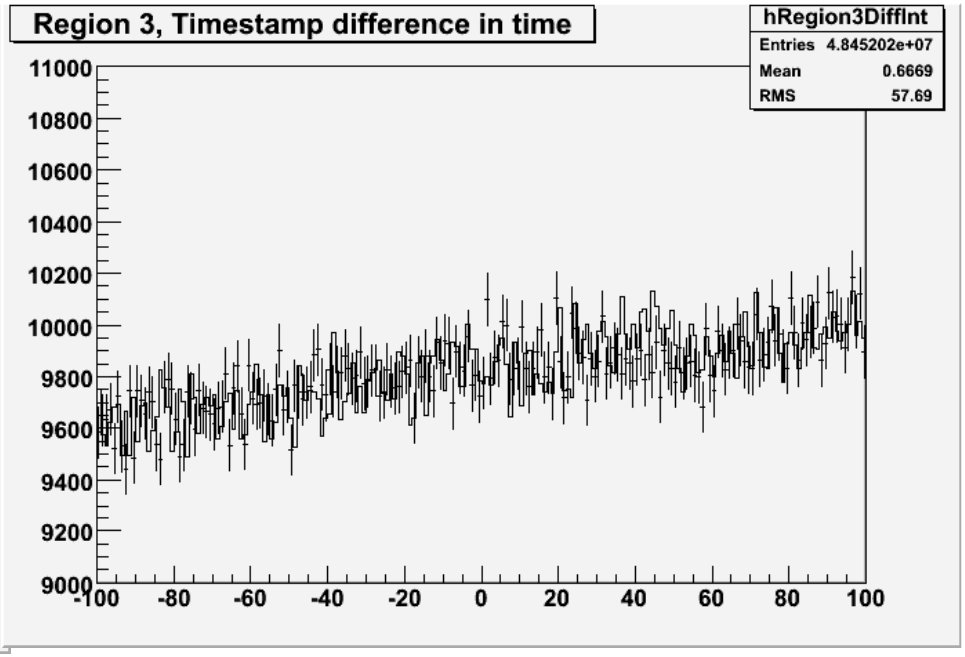




- Signal seen in both region 0 and 1; ~10k and 20k respectively
- ~6M tags total; number of expected hits per region due to source ~1M
- Efficiency of O(1%)? Need to check:
  - Threshold, effect of full memories, timestamp cut, etc.
  - Operating point unlikely to be optimal

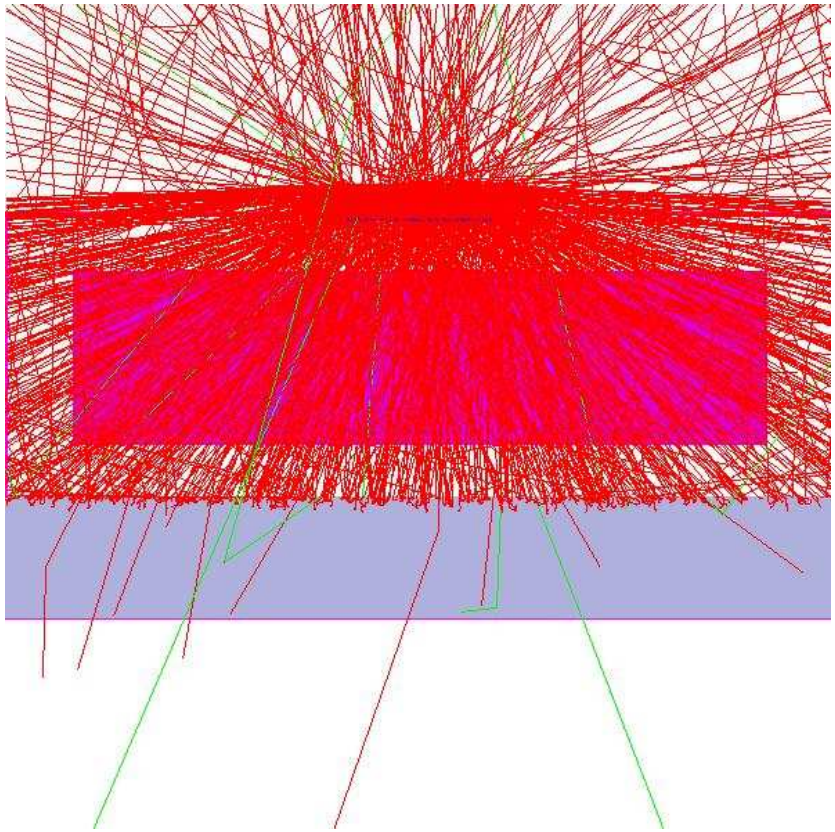
The samplers show no signal

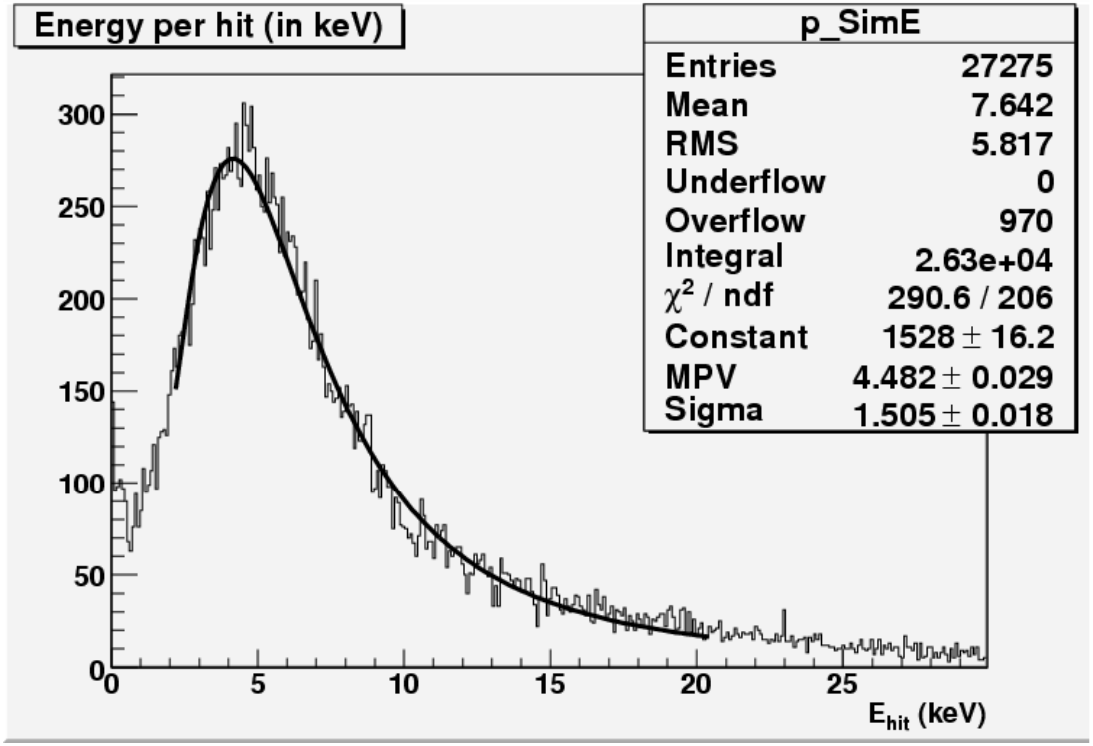




- Nothing at all visible
- May be completely insensitive?

Simulation is running





- Slow electrons deposit a MPV of 4.5keV, compared with a MIP MPV of 3.2keV; source will give ~40% larger signal

Firmware priority list after last meeting

- 1) Sensor configuration data: All writable and readable in same job as sensor bunch train readout. Done
- 2) USB_DAQ configuration data: Make all data readable (including PMT settings?). Are all input levels to the sensor controllable? Add all future settings needed. Done
- 3) Sensor bunch train data: Read status of all sensor output levels. How many are there? Just overflow flags? Done
- 4) Time tagging: Record all 10 inputs, with default (unconnected or no hit) at zero. Add zero suppression. Done
- 5) Laser control: Output control and signal timings.
- 6) Multiple USB_DAQs: Synchronisation.
- 7) External triggering: Tag using external inputs to flag good bunch train. If not good, then flush sensor memory and restart bunch train. Need to have limit on number of attempts. How to set "good" condition in general
- 8) Time tagging and output signals: Move to x8 of bunch crossing frequency.
- 9) External triggering: External signal to start bunch train.
- 10) Power pulsing: Time-controlled power up/down as part of bunch train sequence. Must be time controllable and selectable.
- 11) Bunch crossing frequency: Increase from 2.5MHz to 5MHz. Can this be configurable; if so, can it go to 1.25MHz, etc?
- 12) USB data transfer limitation: 0.5MBytes/s currently. Where is this limited? Can we remove the zero-channel hits (without filling memory)?

Current firmware capability

- Laser system
  - Can run in unsynchronised mode, i.e. fire laser and take bunch trains independently
  - Wait for laser time tag to appear in bunch train readout (like PMT)
  - Bunch train ~2ms, laser rate ~50Hz; one tag per ~10 bunch trains
- Cosmics
  - Run like source tests; look for time tag but require coincidence of two PMT hits. Rate will be low; ~0.01Hz?
  - Sensor hits will be as clean (or unclean) as source case when coincidence seen
  - No inter-USB_DAQ board synchronisation yet; can only run one layer for now