

Default zero suppression in simulation

DUNE 35-ton simulation, reconstruction and analysis

David Adams

BNL

November 18, 2015

Updated 15:50 EST

Introduction

I have been working on zero suppression (ZS)

- Have simulation of the ZS planned for 35-ton running
 - See description of this and current simulation default at https://cdcvs.fnal.gov/redmine/projects/35ton/wiki/Data_compression_and_zero_suppression
- Working on new DetSim module that can use the new ZS sim
 - Will allow user to plug in different services for ZS
 - And for multi-channel ZS and compression
 - Also developing modules to reproduce current behavior
- For more info, see my talk at Monday's DAQ meeting:
 - <https://indico.fnal.gov/conferenceDisplay.py?confId=10621>

I have been validating the new code

- Found the current default ZS does not have much effect
 - With our current noise model
 - Subject of this talk—see following slides

DetSim configuration (1)

I use the fcl at right

- I believe this is our current default for simulation
- Resulting fcl parameters for DetSim are on the following page

```
#include "services_dune.fcl"
#include "singles_dune.fcl"
#include "largeantmodules_dune.fcl"
#include "detsimmodules_dune.fcl"

process_name: SinglesGen

services:
{
  # Load the service that manages root files for histograms.
  TFileService: { fileName: "mu35t_hist.root" }
  TimeTracker: {}
  RandomNumberGenerator: {} #ART native random number generator
  user: @local::dune35t_simulation_services
}

#Start each new event with an empty event.
source:
{
  module_type: EmptyEvent
  timestampPlugin: { plugin_type: "GeneratedEventTimestamp" }
  maxEvents: 1 # Number of events to create
  firstRun: 1 # Run number to use for this file
  firstEvent: 1 # number of first event in the file
}

# Define and configure some modules to do work on each event.
# First modules are defined; they are scheduled later.
# Modules are grouped by type.
physics: {
  producers: {
    generator: @local::dune35t_singlep
    largeant: @local::dune35t_largeant
    daq: @local::dune35t_simwire
    rns: { module_type: "RandomNumberSaver" }
  }
}

#define the producer and filter modules for this path, order matters,
#filters reject all following items. see lines starting physics.producers below
simulate: [generator, largeant, daq, rns]
```

DetSim configuration (2)

```
daq: {  
  CollectionPed: 500  
  CollectionPedRMS: 0.01  
  CompressionType: "ZeroSuppression"  
  DriftModuleLabel: "largeant"  
  FractHorizGapUCollect: 0.1  
  FractHorizGapUMiss: 0.8  
  FractHorizGapVCollect: 0.1  
  FractHorizGapVMiss: 0.8  
  FractHorizGapZMiss: 0.8  
  FractUUCollect: 0.5  
  FractUUMiss: 0.2  
  FractUVCollect: 0.1  
  FractUVMiss: 0.2  
  FractVUCollect: 0.5  
  FractVUMiss: 0.2  
  FractVVCollect: 0.1  
  FractVVMiss: 0.2  
  FractVertGapUCollect: 0.1  
  FractVertGapUMiss: 0.8  
  FractVertGapVCollect: 0.1  
  FractVertGapVMiss: 0.8  
  FractVertGapZMiss: 0.8  
  FractZUMiss: 0.2  
  FractZVMiss: 0.2  
  InductionPed: 1800  
  InductionPedRMS: 0.01  
  LowCutoffU: 7.5  
  LowCutoffV: 7.5  
  LowCutoffZ: 7.5  
  NearestNeighbor: 25  
  NeighboringChannels: 3  
  NoiseArrayPoints: 1000  
  NoiseFactU: 0.332  
  NoiseFactV: 0.332  
  NoiseFactZ: 0.332  
  NoiseModel: 1  
  NoiseOn: 1  
  NoiseWidthU: 62.4  
  NoiseWidthV: 62.4  
  NoiseWidthZ: 62.4  
  PedestalOn: "false"  
  SaveEmptyChannel: "true"  
  SimCombs: "false"  
  SimStuckBits: "false"  
  StuckBitsOverflowProbHistoName:  
  "pCorrFracOverflowVsInputLsbCell"  
  StuckBitsProbabilitiesFname:  
  "ADCStuckCodeProbabilities35t/  
  output_produceDcScanSummaryPlots_20150827_c  
  oldTest_0p1to1p4_step0p0010.root"  
  StuckBitsUnderflowProbHistoName:  
  "pCorrFracUnderflowVsInputLsbCell"  
  ZeroThreshold: 5  
  module_type: "SimWireDUNE35t"  
}
```

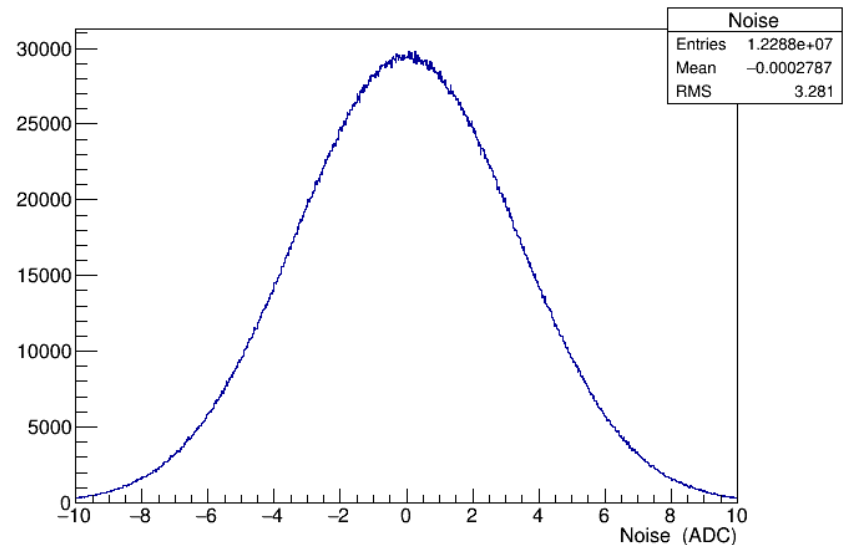
Noise

Histograms show noise

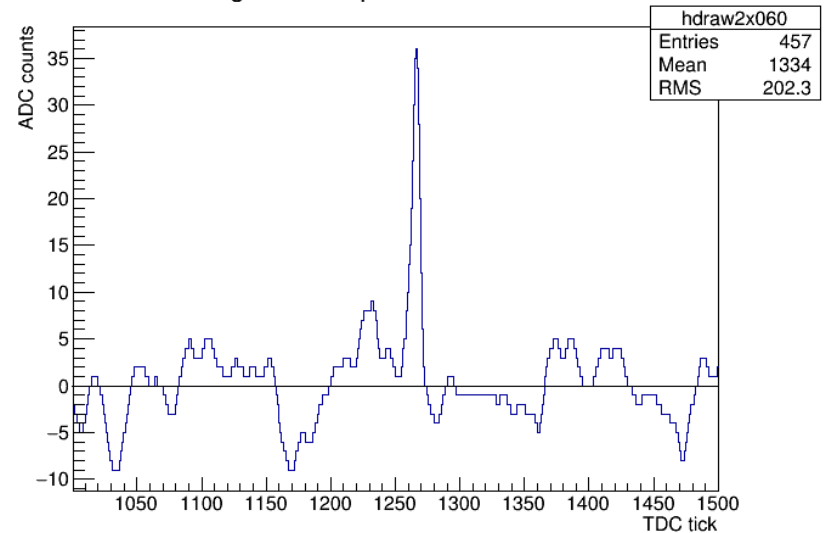
- Top is noise histogram from DetSim module
- Bottom is part of ADC spectrum from one wire

Comment

- Signal is evident on wire
- Noise clearly has time correlations
- Noise often dips above or below threshold of 5
- No ZS is evident
 - Presumably because of ZS channel window



Raw signals for apa0z2 event 1 channel 60



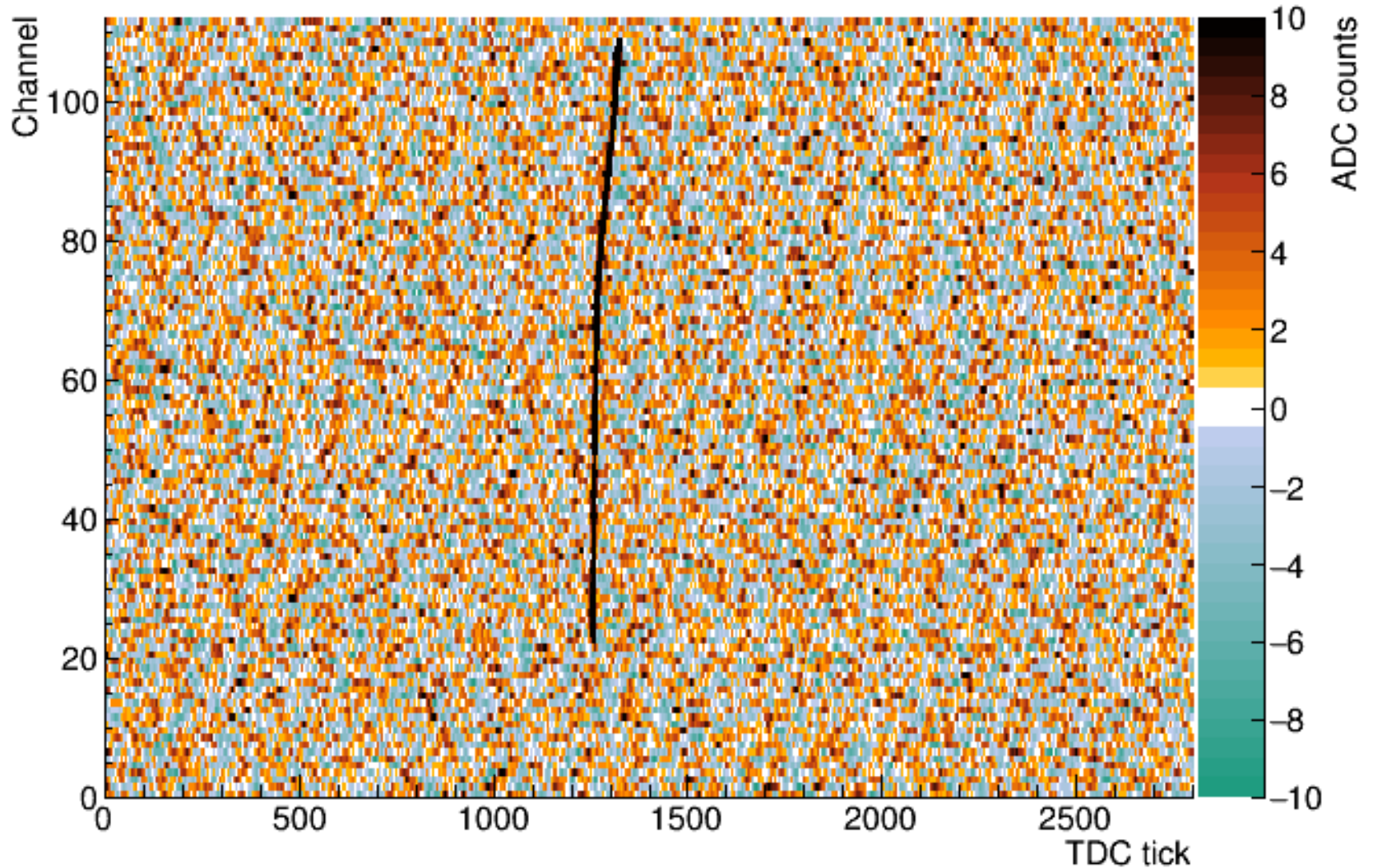
Zero suppression

Following slides show 2D ADC spectrum for one plane

- Channel number vs. TDC tick
- Four plots:
 - No ZS
 - Default channel window sizes (NeighboringChannels) Nch = 3
 - Nch = 1 (only keep adjacent channels)
 - Nch = 0 (no neighboring channels)
- Note distortion for small channel numbers when ZS is applied
 - Defect in ZS or unpacking code?

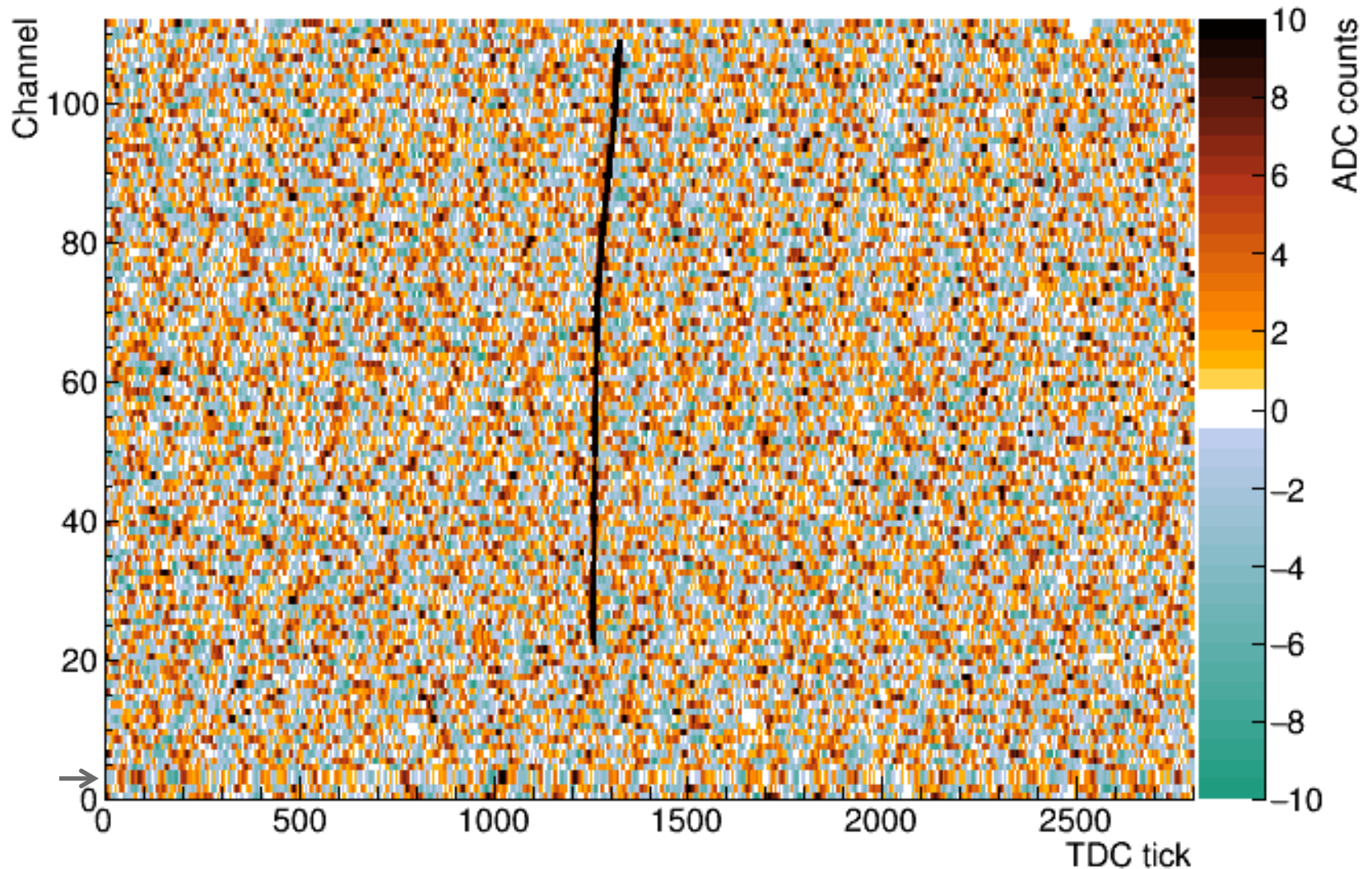
ADC spectrum without ZS

Raw signals for apa0z2 event 1



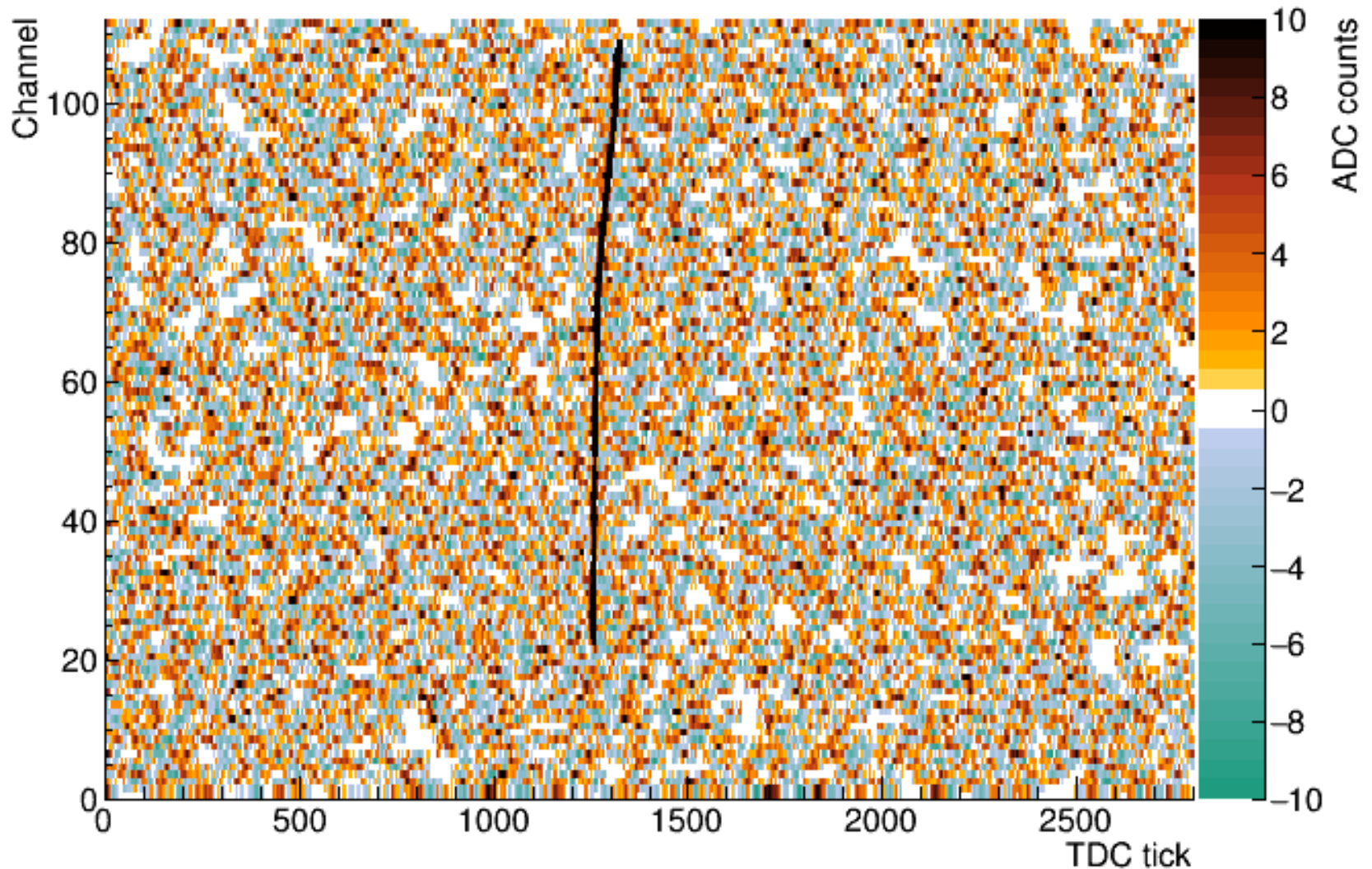
ADC spectrum for ZS channel window of 3

Raw signals for apa0z2 event 1



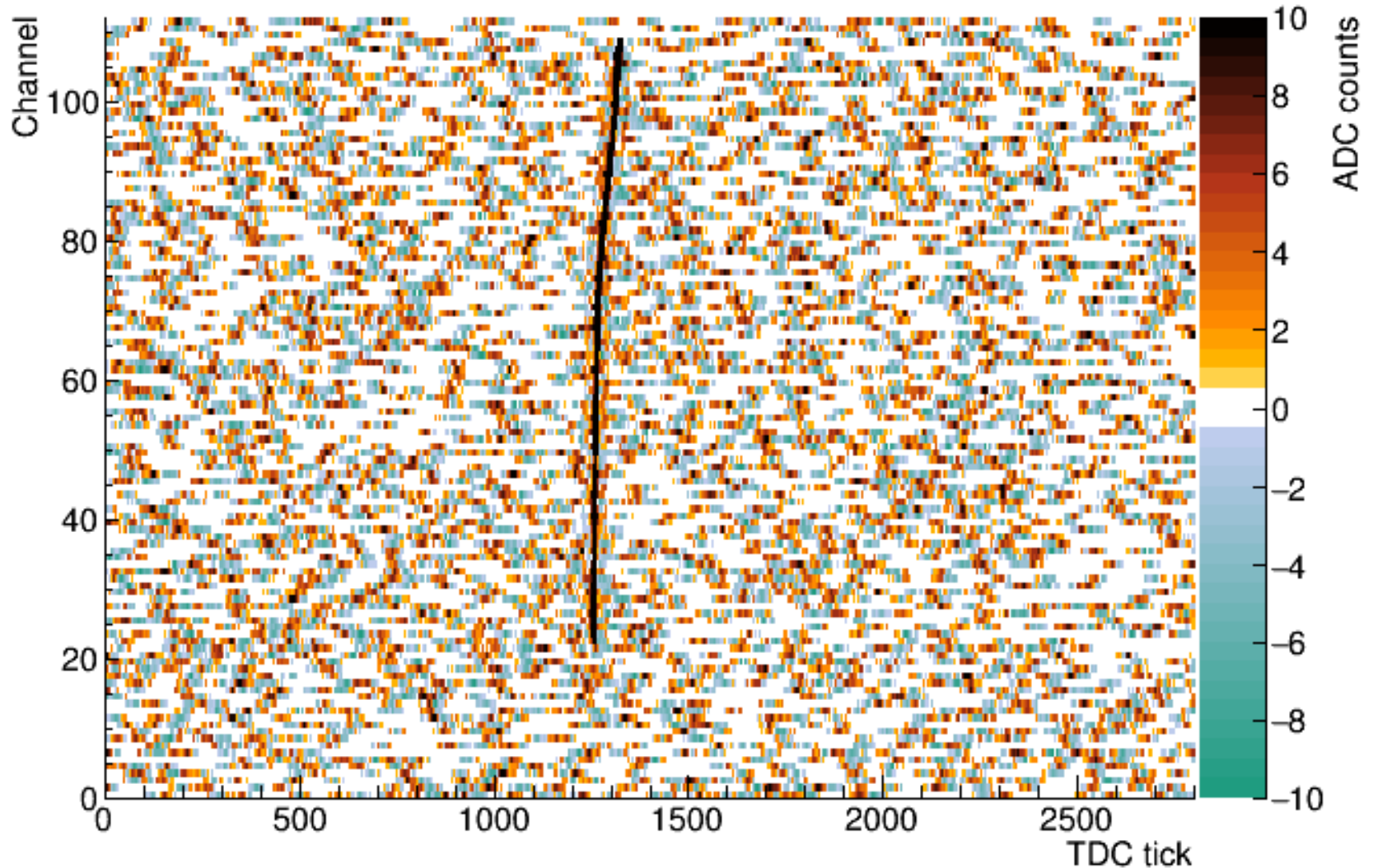
ADC spectrum for ZS channel window of 1

Raw signals for apa0z2 event 1



ADC spectrum for ZS channel window of 0

Raw signals for apa0z2 event 1



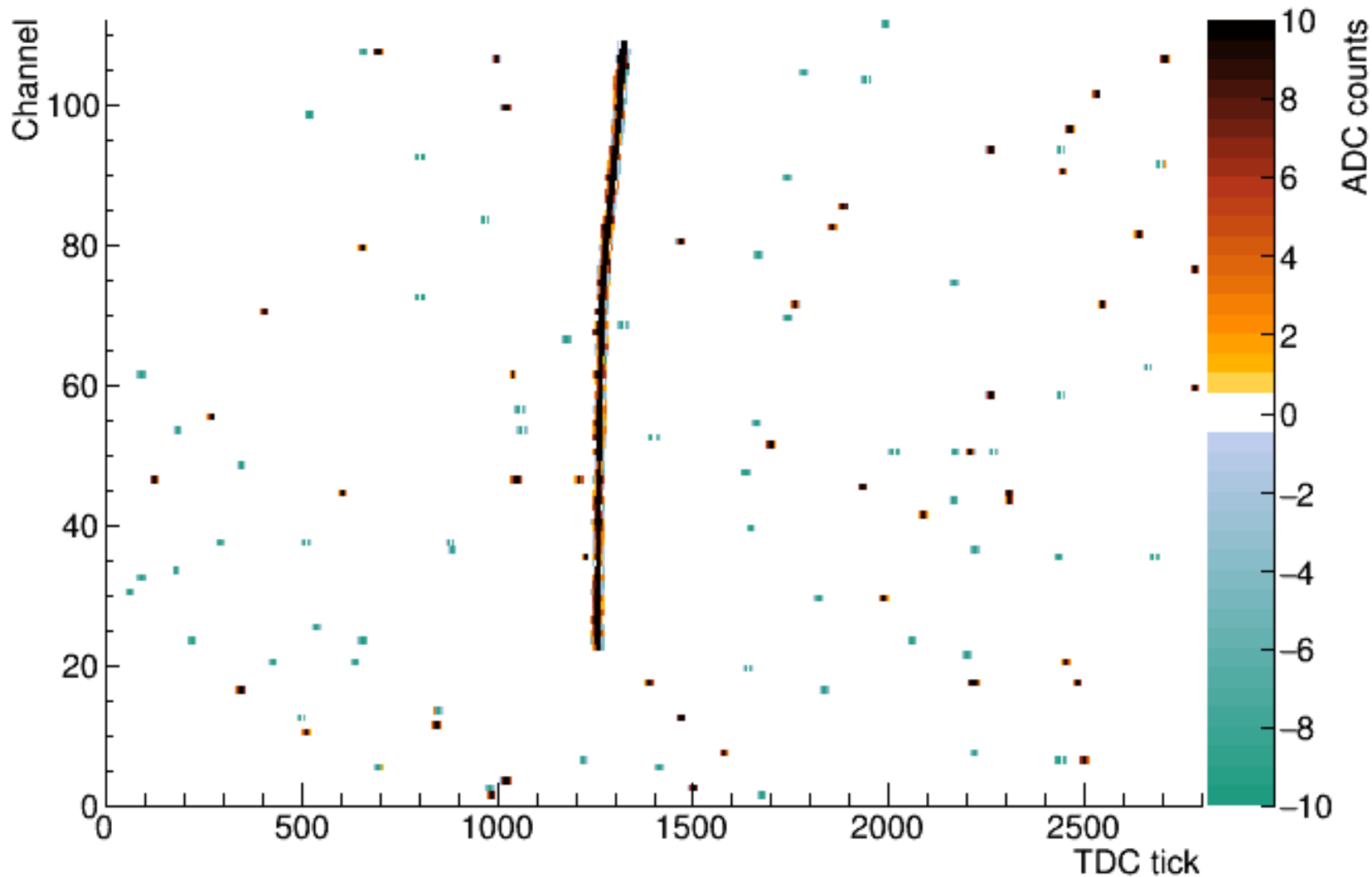
Zero suppression comments

Comments

- Current default suppresses almost nothing
- Ignoring neighbors entirely only suppresses about half
- For this signal/noise simulation, we should
 - Use one or better zero neighboring channels
 - There is no plan for more than zero for 35 to DAQ
 - Decrease tick window: maybe 25 → 10
 - Raise threshold: 5 (1.5σ) → 10 (3σ) or more
 - See next two plots for the effect of these parameters
- Or don't bother to include ZS in simulation
- Or switch to simulation of ZS being developed for the DAQ
 - Available soon

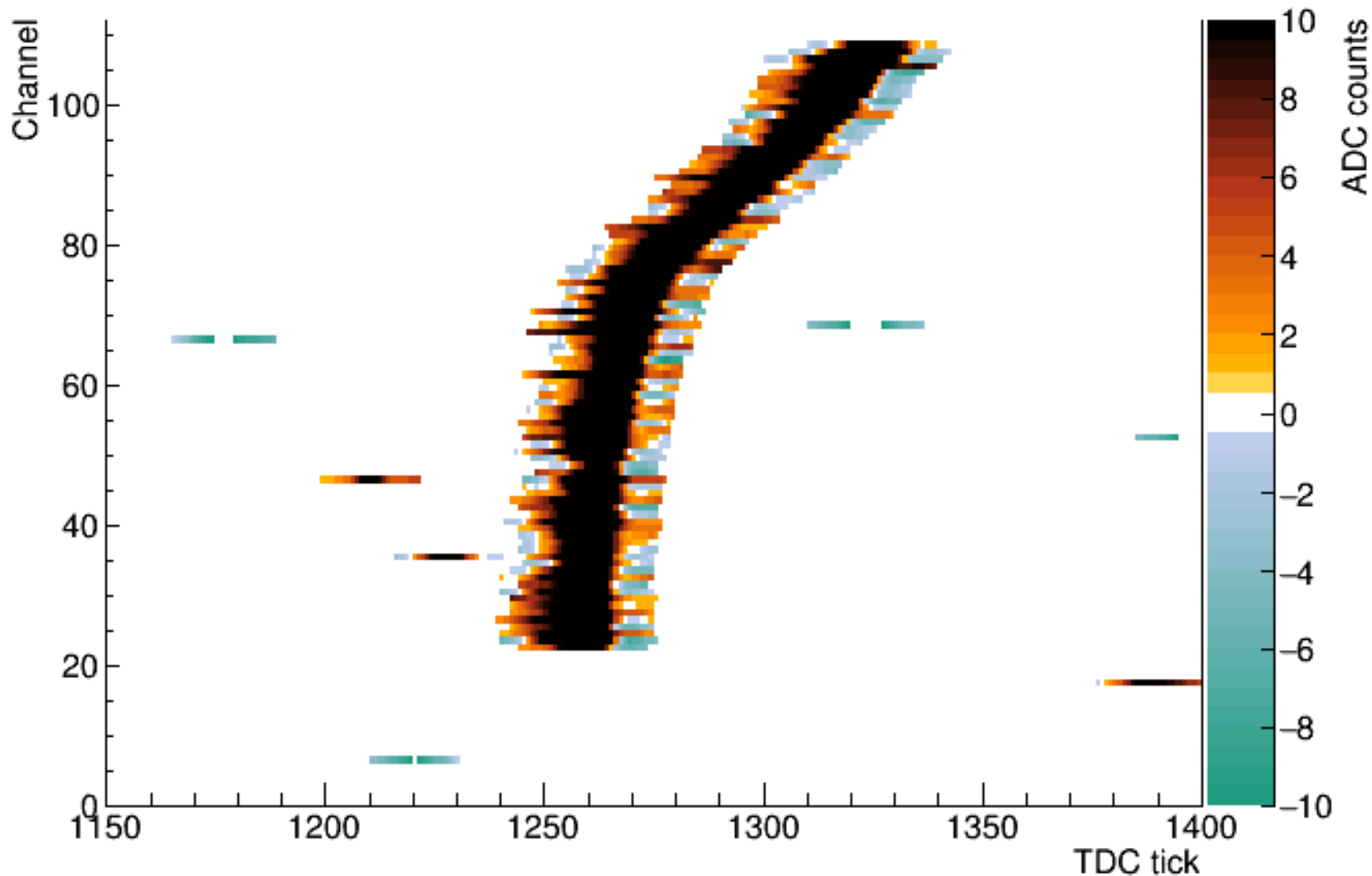
ADC spectrum for ZS Nch=0, Ntick=10, th=10

Raw signals for apa0z2 event 1



ADC spectrum for ZS Nch=3, Ntick=10, th=10

Raw signals for apa0z2 event 1



Current status of ZS development

New code is in github:

- <https://github.com/dladams/dunezs>

This package includes

- Interface for ZS
- ZS implementations for legacy (window) and 35 ton online sim
- Service wrappers for ZS interface and implementations
- Compression which replaces suppressed values with zero
 - Plan to add interface and implementation with current block packing
- New SimWire module: SimWireDUNE
 - Copied from SimWireDune35t
 - ZS/compression section replaced with call to ZS and compression services

Validation

- I ran 5 events and compared current and new implementations
 - With my preferred values for ZS
- 17/80 planes, 21/29M bins differ.
 - I am investigating. One example is on the following page.

ZS development discrepancy

Plot at right shows discrepancy

- Plot is the new code
- Original does not have hole at channel 1287
- But it is 11 channels away from the nearest bins with <-10 and >10 and so should be suppressed. No?
- Or I don't understand the intent of the current algorithm

