ADC simulation (and other) tools

LArSoft

David Adams
BNL
May 9, 2017

Introduction

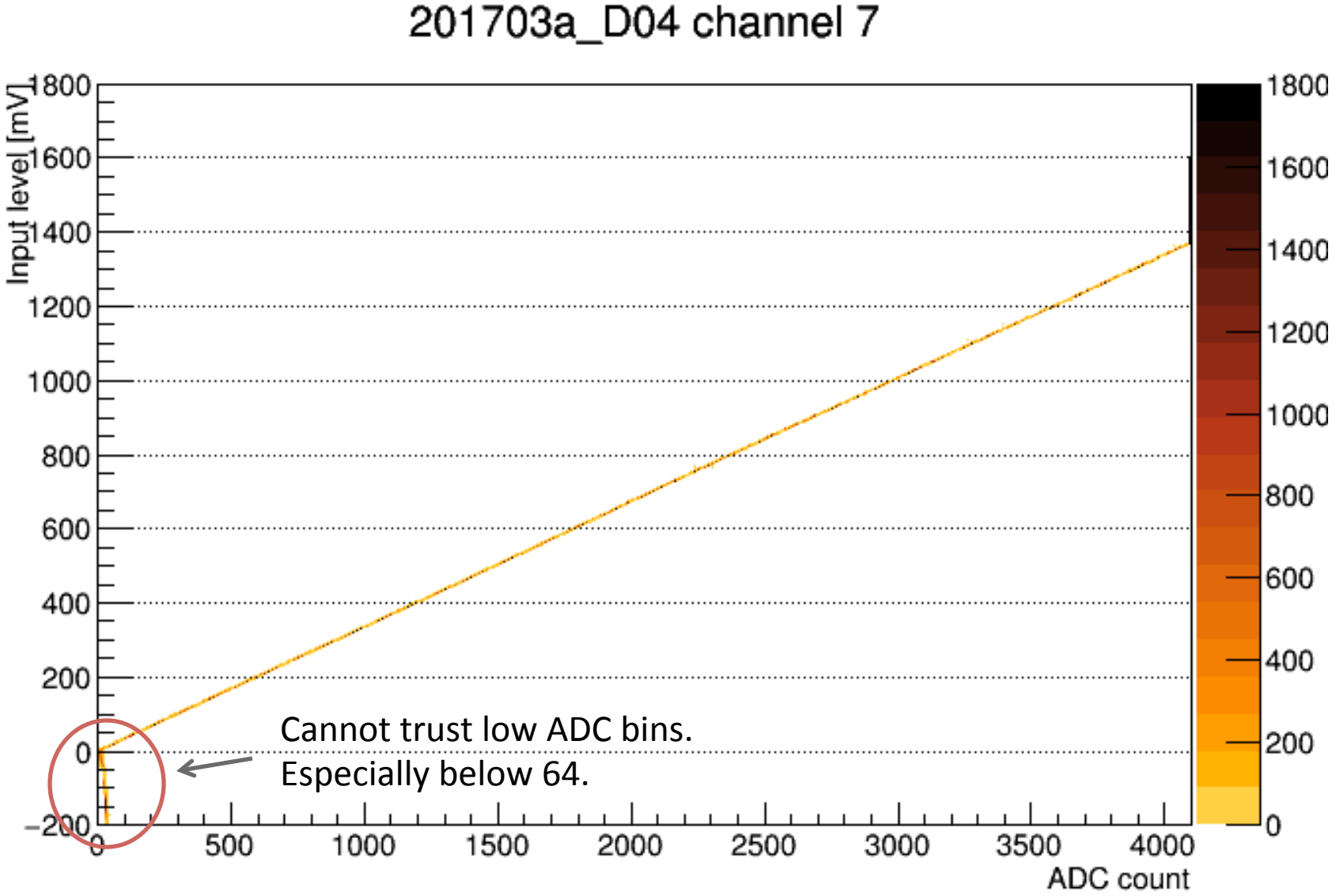
I have been looking at ADC test data taken at BNL

- For the P1 ADC version we will use in protoDUNE
- Performance is far from ideal but much better than 35t
- See following slides and talks at DUNE cold electronics and others

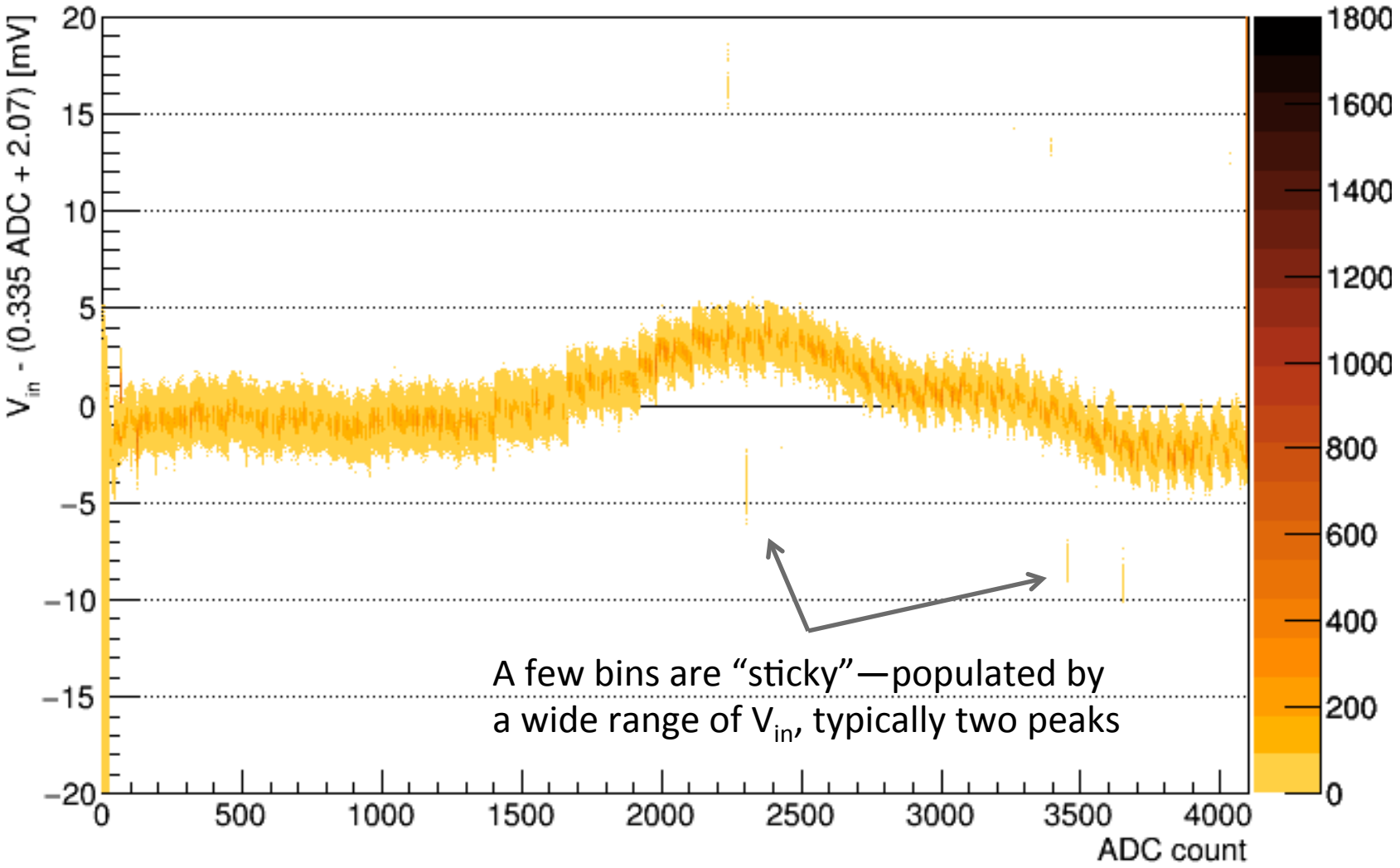
Like to include realistic ADC in DUNE simulation

- I am working on tools to provide this
  - Input: input voltage, channel #, time/event
  - Output: ADC count/bin (i.e. in range 0 – 4095)
- Subject of this talk are those tools
  - And tools in general
- I would like these tools to usable in other contexts
  - Other experiments: SBND, …
  - Outside art framework: Root macros, standalone quick simulations
- Details follow

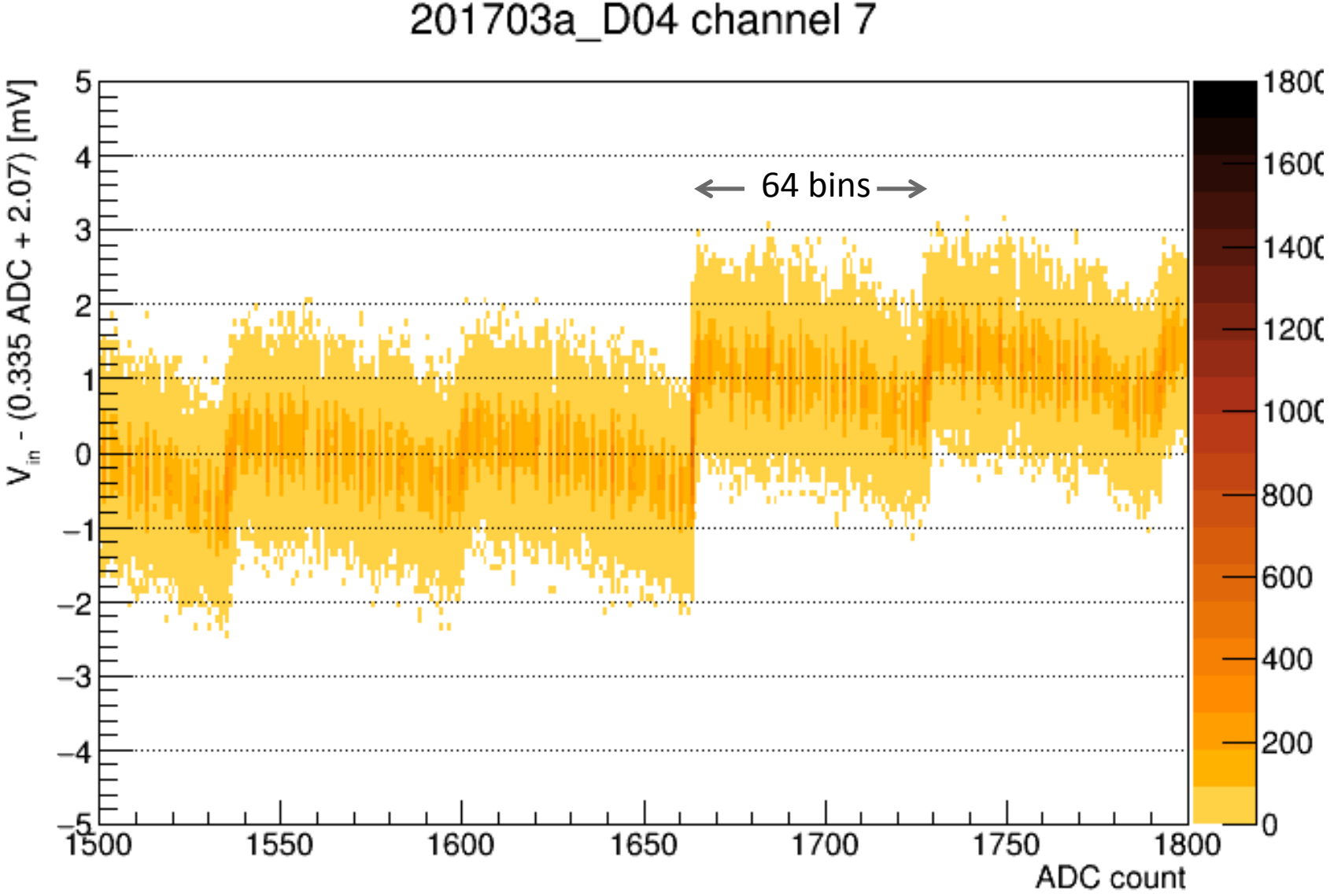
Example P1 inverse response



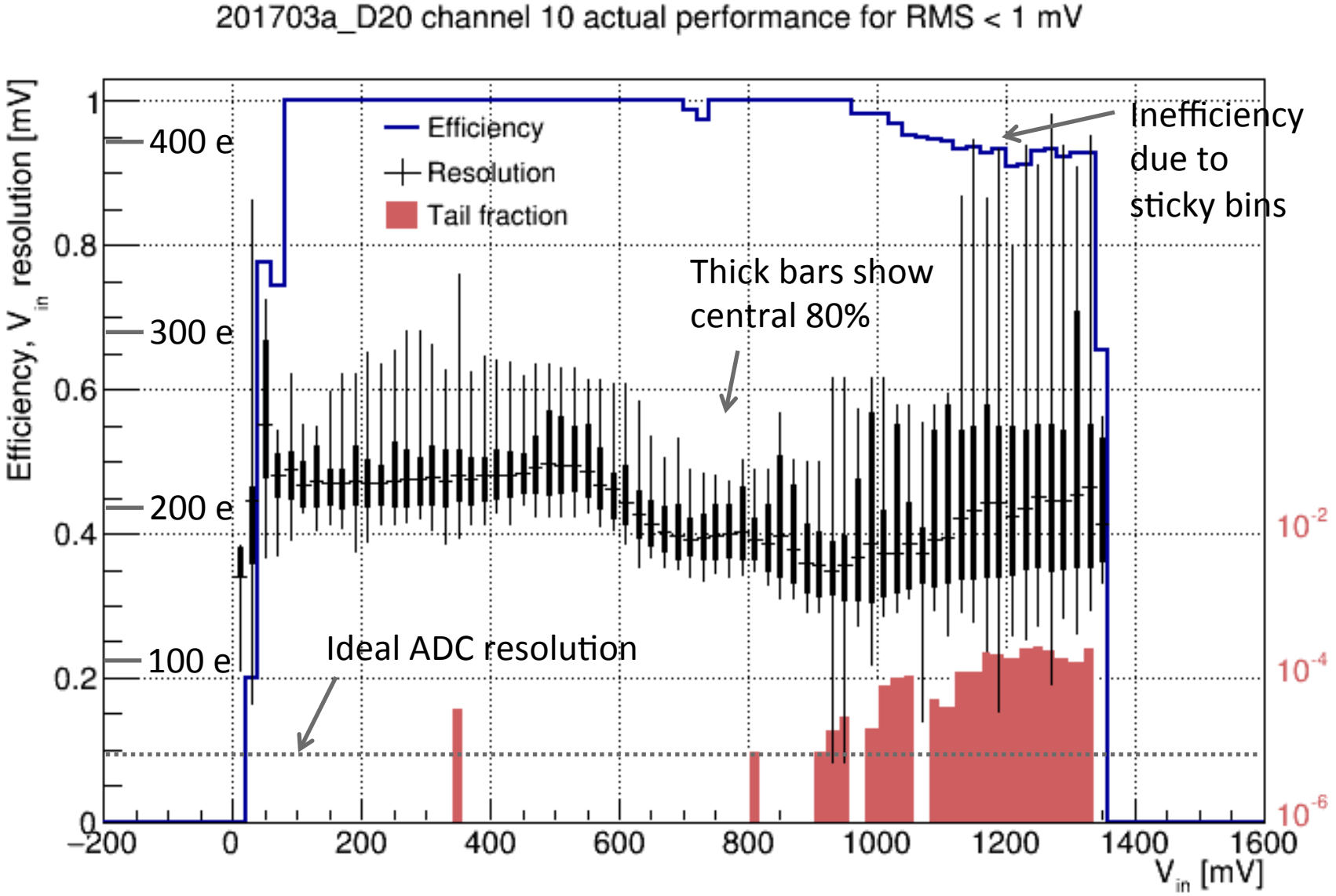
Example P1 linear fit residual



Example P1 linear fit residual zoomed



Example P1 performance summary plot



Software structure 1

Requirements

- Support multiple simulation strategies
  - Ideal ADC
  - Any one of the tested ADC chips
  - Detector mapping to many ADC chips
  - And more (e.g. dual ADC ideas)
- ADC simulators should be easily usable by all interested parties
  - Both inside and outside art event-processing framework
- Easily-shared, named configurations
  - E.g. “adcsim_mar2015_D05” run in two places will give the same results
- Minimal compile and linker dependencies
  - Should be easy to plug in a different ADC simulations
- Expect similar requirements for many other simulation, reconstruction and analysis tools

Software structure 2

Adopted solution

- ADC simulators are constructed as art class tools
  - See https://cdcvs.fnal.gov/redmine/projects/art/wiki/Guide_to_writing_and_using_tools
- ADC simulator has a base interface AdcSimulator
  - Clients access simulators via this interface
  - See dunetpc/dune/DuneInterface/AdcSimulator.h
    - Hope to move this into larsoft soon
    - In dedicated package so clients need not depend on any of rest of larsoft
- Concrete simulators inherit from this base
  - Include art CPP macro to register tool
  - Provide ctor from FCL for configuration
- Tool manager allows clients to access tool configurations
  - By interface type (e.g. AdcSimulator) and name (e.g. adcsim_ideal)
  - FCL configuration file maps names to configuration
  - Configuration includes the name of concrete type and all parameters required by the ctor for that type
  - See dunetpc/ArtSupport/DuneToolManager.h
    - Hope to replace/supplement this with art or larsoft tool manager

ADC simulation tools

Plan the following concrete ADC simulators

- Ideal ADC simulation
  - Already implemented—see following slides
  - See IdealAdcSimulator in dunetpc/dune/DetSim/Tool
    - Expect to move this to dedicated larsoft package
- Simulation based on results from test stand measurements
  - For any one ADC chip measurements
  - Data provide an ADC count pdf for any input voltage
  - Simulator would randomly select from the pdf
    - Begin without time correlations
- Many-chip simulator
  - Passes call to single-chip simulator based on channel map
- Easy for others to provide additional implementations

DUNE detector simulation

DUNE DetSim is already modified to use the AdcSimulator

- Tool is accessed in the SimWireDUNE module
  - Used for DUNE 35t, protoDUNE, FD, …
  - Probably could/should be renamed and moved to larsoft for use by other experiments
- Previously ideal ADC simulation was embedded in the module
- Now module looks for and uses an AdcSimulator
  - Tool name is a FCL parameter for the module
  - If blank, old simulation is used with (lots of) warning messages

ADC gain

- At present, the signal-shaping service (SSS) converts collected charge to floating ADC counts
  - So ADC simulator must have a gain of one
- I would prefer to have SSS convert charge to ADC input voltage
  - And let ADC simulator convert voltage to ADC count
  - If no objection, I will make this change

SW packaging

How to organize the code?

- Minimize dependencies
  - So clients don't have to install (or compile or port) unneeded SW
- But don't want too many packages repositories
  - Confusing for clients
- These are competing requirements

Proposal on following page

SW packaging (2)

Proposal: three components

- Interface packages only depend on data classes
  - One or more with no dependencies (define their own transient data)
  - Other(s) depending on lardataobj (larsoft data classes)
    - Could break that dependency if containers held pointers instead of objects
  - Tool clients will have compile-time dependency on these packages
- Tool manager package
  - Might go into ArtUtilities
  - Used by tool clients (including other tools) to find tools
    - Manager (or its interface) must be shared by all those sharing tools
- Tool implementations: many packages
  - At all levels: larsoft, experiment, analysis group, individual
    - May also want to split by realm: simulation, reco etc.
  - Depend on the above and little bit of art
  - Clients do not have build dependency (i.e. compile or link) on these tool implementation packages
    - Instead dynamic link

Summary/Conclusions

ADC simulation tools are being developed

- Ideal ADC simulator is available now
- P1 simulator based on test stand measurements coming soon
- Plan to add tool that selects between multiple chips

ADC simulator follows new tool pattern

- Propose to use this for future SW development
- Define interface for each type of tool
- Provide one or more implementation of the interface as art class tools
- Define named tool configurations in FCL
- Client uses tool manager to find named tool configurations
  - Typically name is a FCL parameter of the tool client
- Packaging
  - Dedicated packages for tool interfaces
  - Package to hold tool manager
  - Packages for concrete tools (at all levels)

Extras

Validation of ideal ADC simulator

Tool manager

Validation of ideal ADC simulation tool

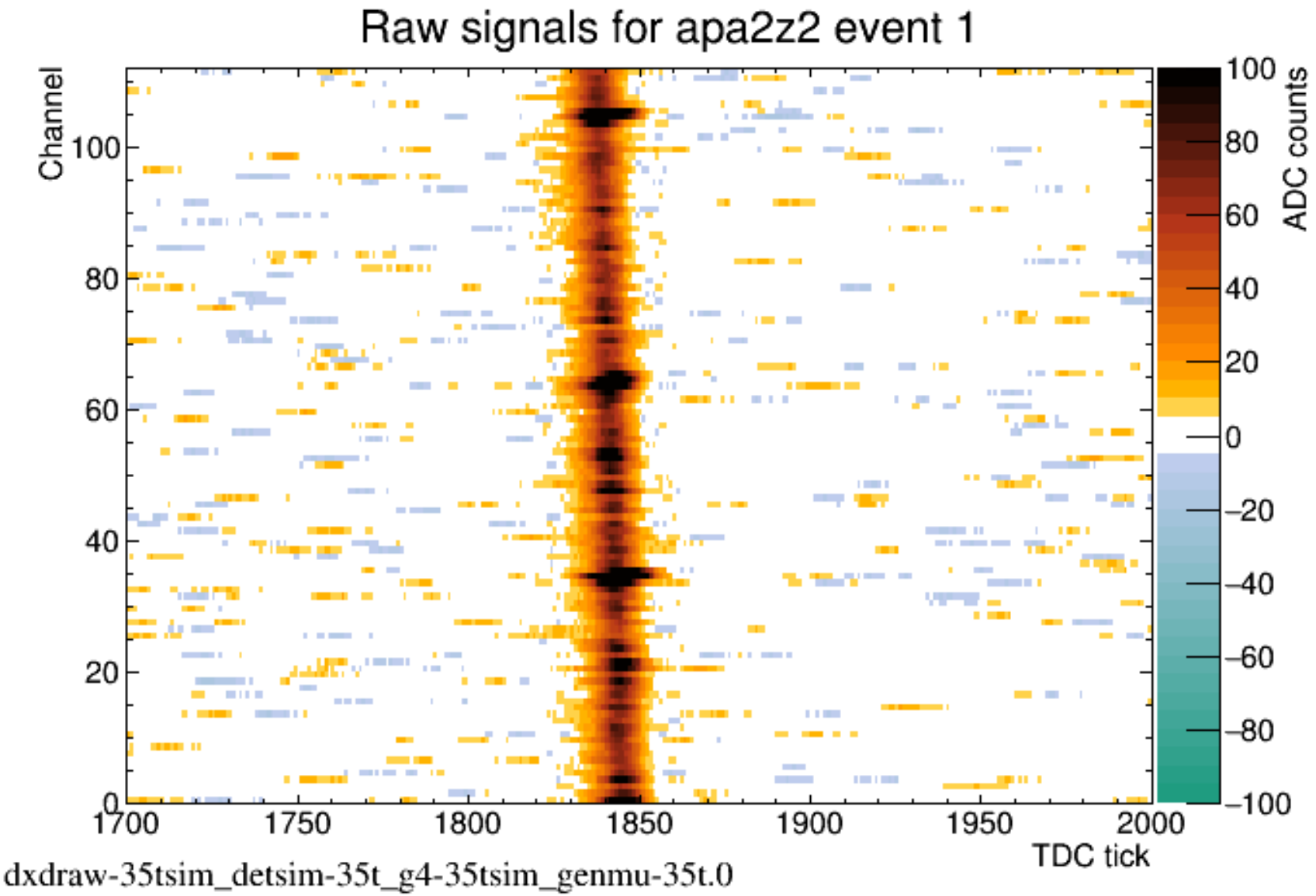
Following slides compare old and new ADC simulation

- Old is the code embedded in SimWireDUNE module
- New is the updated module using IdealAdcSimulator (adcsim_ideal)
- Results show for three planes in one event
- All for the 35t standard simulation of a single muon
- No visible differences

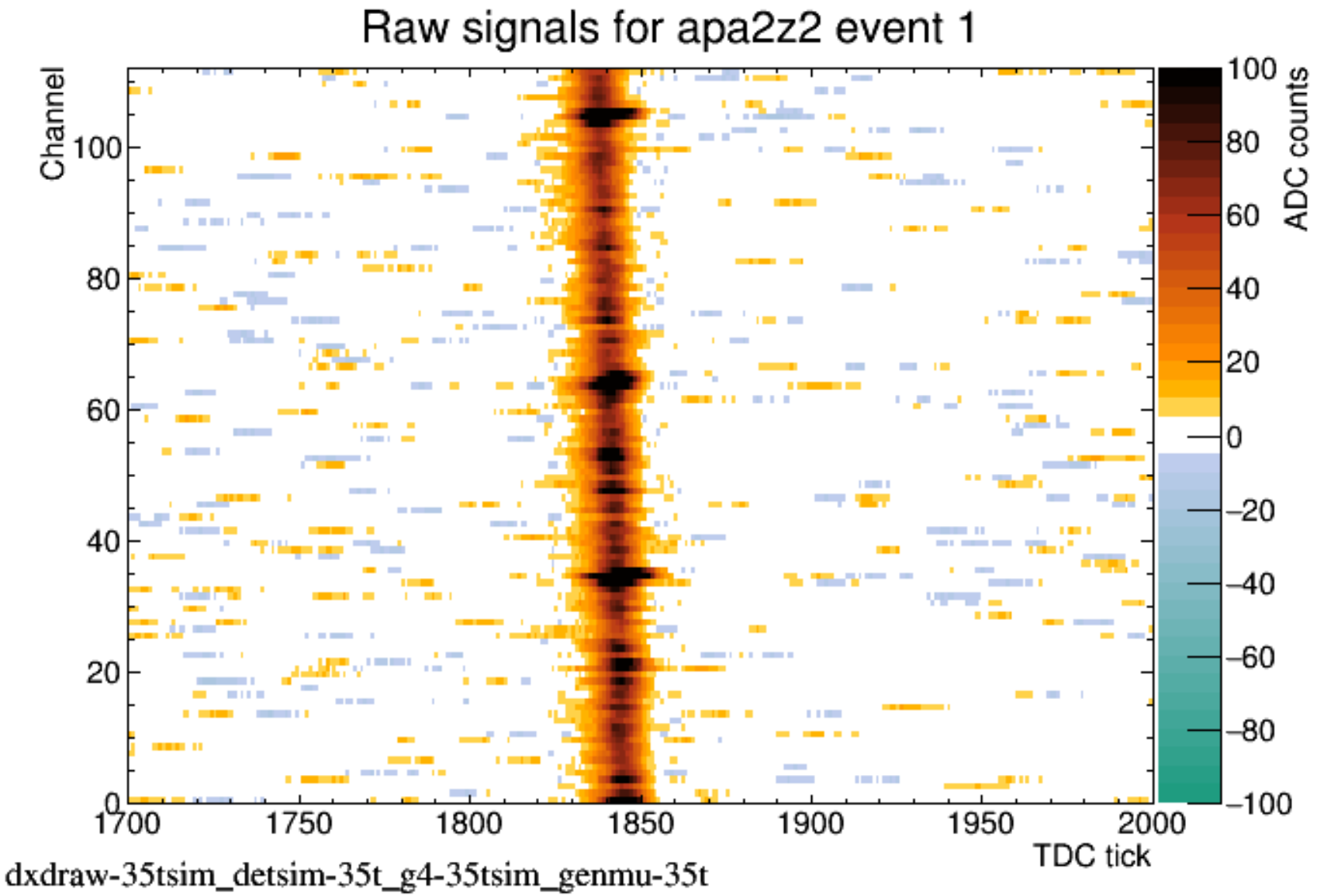
Direct check

- I also compared the old and new values for 10 events in the code and saw no differences
- For both 35t and FD126 geometries

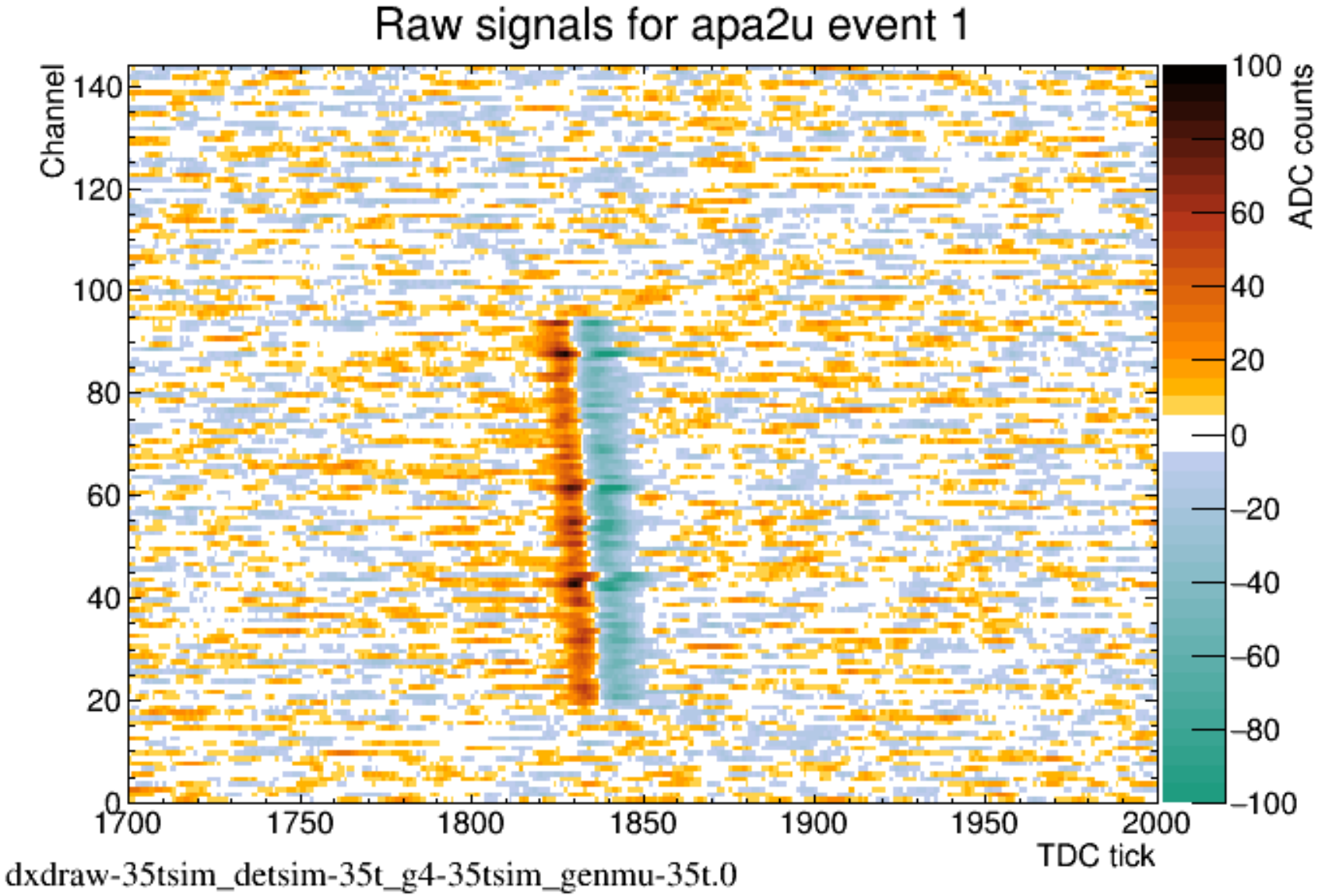
Old simulation: 2z2



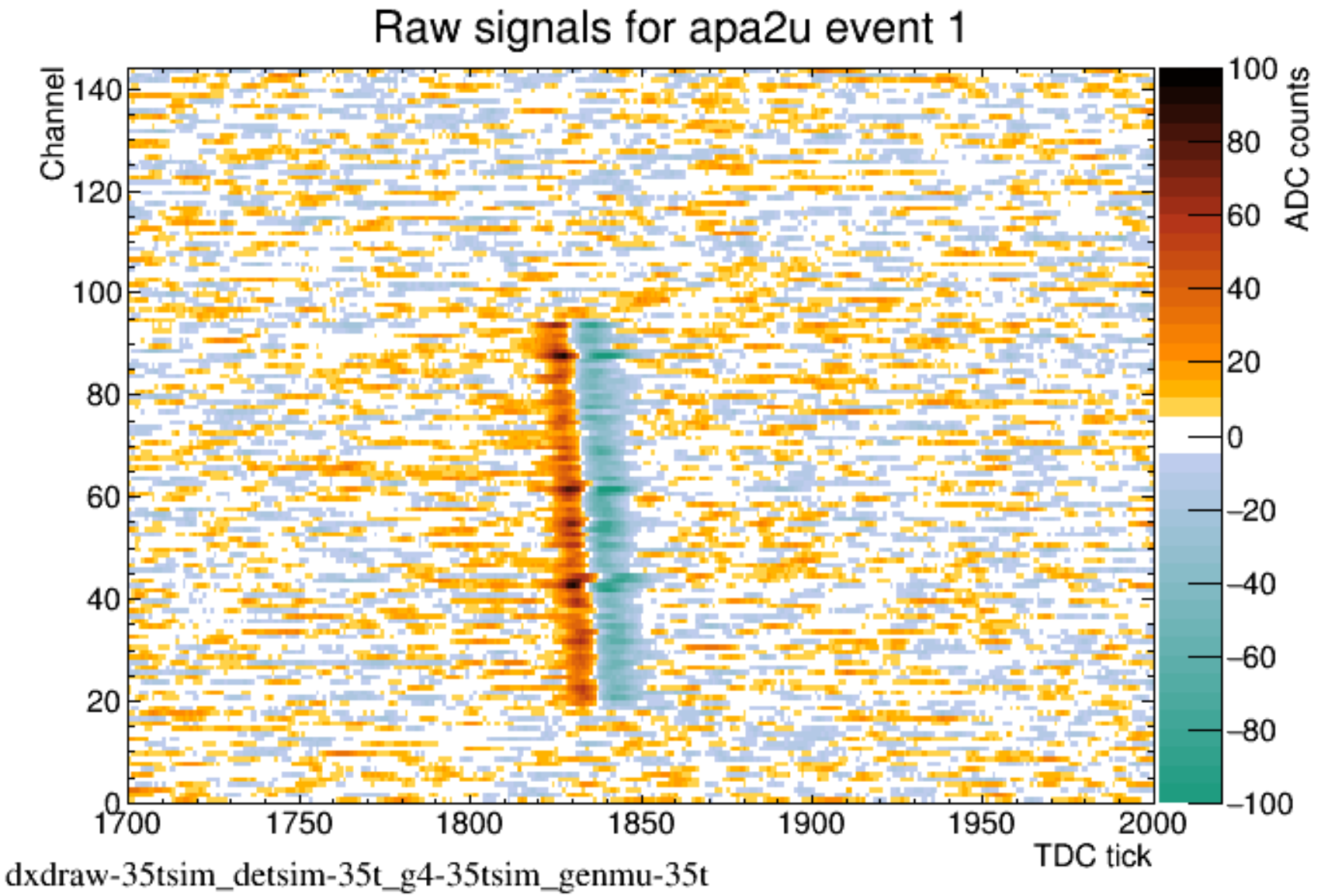
New simulation: 2z2



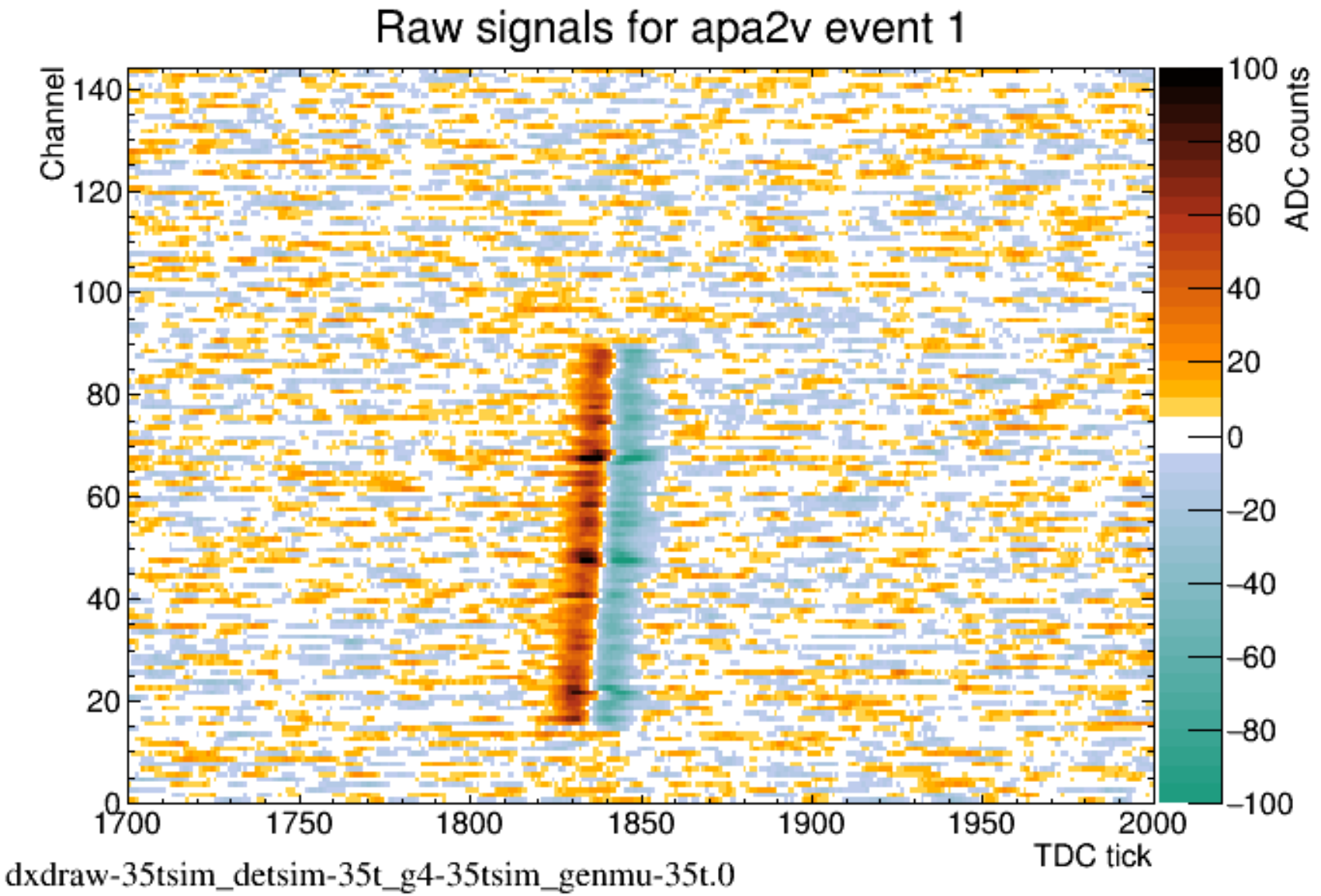
Old simulation: 2u



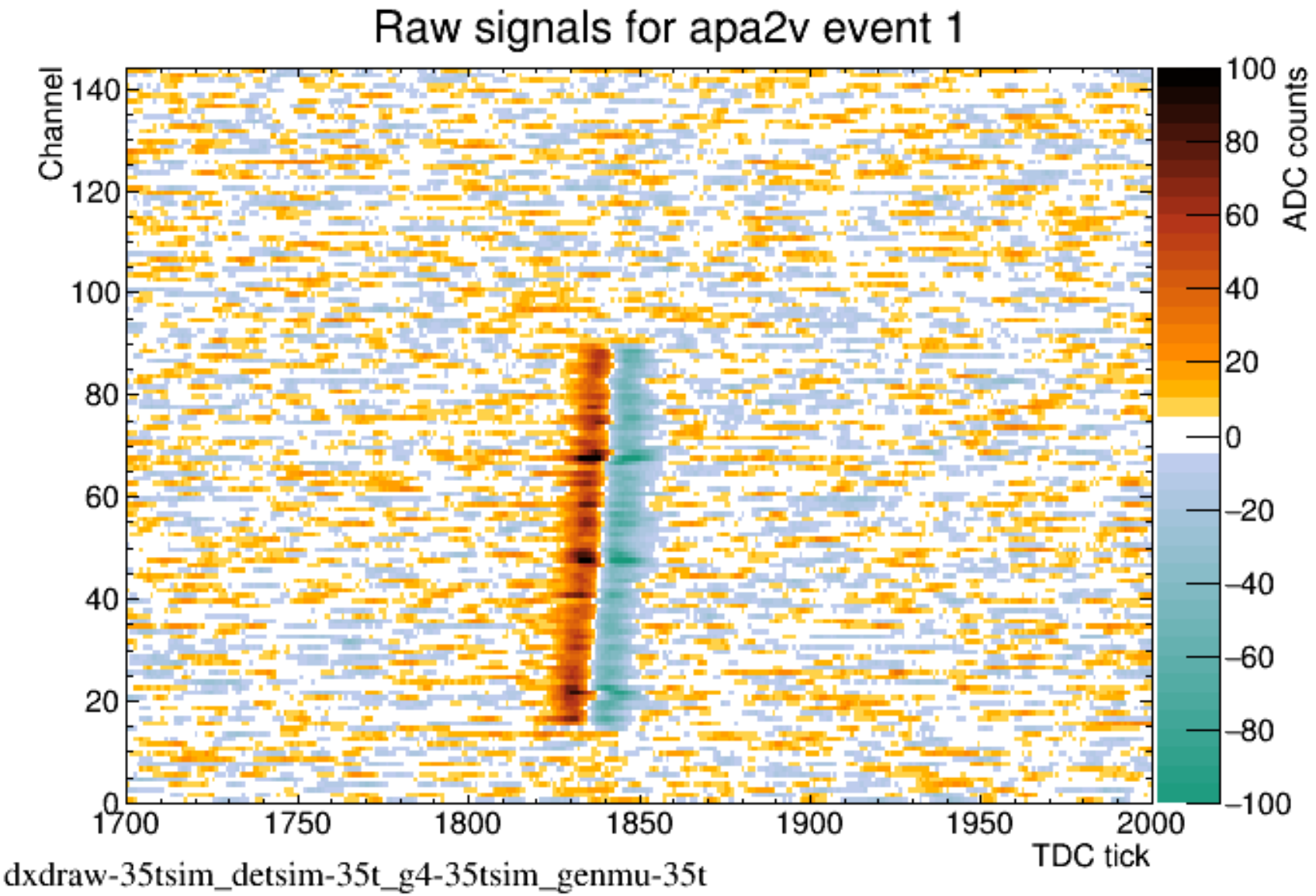
New simulation: 2u



Old simulation: 2v



New simulation: 2v



Tool manager

What does tool manager tool do?

- Provides access to tool specified by interface type and name
  - Expect tool configuration to have a unique name
- Mapping from name to configuration taken from FCL
  - tools.MyName = {…}
  - Later could be other sources or languages
- Envision different types of tool access
  - Private: client obtains private copy of the tool
    - Client responsible for lifetime management
    - Current art model
    - Current implementation only has this
  - Global: all clients share the same instance
    - E.g. for caching, DB access, sharing information, logging, …
  - Limited sharing
    - E.g. one instance per event or thread