Some Examples using Gallery

Tom Junk

DUNE Computing Tutorial

November 14, 2017

Marc's Gallery Tutorial from August

- I started from the examples there: https://indico.fnal.gov/event/14943/other-view?view=standard
- His example code is available at https://github.com/marcpaterno/gallery-demo
- Clone the repo with:
  git clone https://github.com/marcpaterno/gallery-demo.git
- The setup script in there is a bit much. All you have to do is set up dunetpc and gallery (gallery is not set up by default). I have an alias that does this:
  source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh; setup dunetpc v06_55_01 -q e14:prof; setup gallery v1_05_03 -q e14:nu:prof
- If you need to look up available versions, use ups list -aK+ gallery. ups will warn you if something mismatches.

Reminder of what Gallery is good For

- Escaping the long build time of dunetpc
- Working with "bare" root. I found that just running uncompiled scripts was very fast
- Making plots.
  - To make a good-looking plot, experimenting with label size, positioning, plot color scheme, titles, legends, line and point styles, etc. can take forever.
  - Usual strategy if a plot takes a long time to generate – save it in a rootfile and adjut the cosmetics later.
  - But some "cosmetics" require regenerating the plot!
    - Example: I was experimenting with FFT'ing some raw digits from 3x1x1. Robert Sulej told me that the first few ticks were noisy and not representative. So being able to adjust the tick range without a big rebuild and rerun of a job was a big plus for me.

Reminder of what Gallery is Not Good For

- Adding data products to events (can't do it)
- Accessing *art* Services – though LArSoft has divided its Services into Service Providers and Services so that the provider can be called outside of *art*, so at least some services (those that do not depend on callbacks like discovering run boundaries or new events) can presumably be linked in to gallery executables. I haven't tried. Many services have to be configured, and so you need to instantiate the provider with enough configuration to make it work. e.g. geometry needs GDML.
- accessing provenance or metadata

Some Gallery Scripts

- I had some time to write some not-too-complex scripts to display recob::SpacePoints, and things I can compute from raw::RawDigits, without the need for geometry or even a channel map.
- Marc's examples have more complicated access patterns, such as following associations. Mine don't.
- Look at Marc's demo.cc and analyze.cc and analyze.hh for an example of following associations in gallery.
- Much of DUNE MC is zero-suppressed (some new samples with noise for ProtoDUNE are not). Unpacking code is either in lardataobj (Jonathan Insler's old code) or the new dataprep service. So I ran on 35-ton raw data and 3x1x1 raw data.
- See the scripts in dunetpc/dune/GalleryScripts (these are not built and there's no CMakeLists.txt file for them). You have to check them out of git.

Gallery Scripts

- pdspdisp.C
  - ProtoDUNE-SP display of space points. Colors red the ones that are in a cylinder around the beam axis.
  - Takes the name of an input file – needs to be a reconstructed ProtoDUNE-SP file. Example:

    /dune/data/users/trj/tutorial_aug14_2017/protodune_1gev_reco_withcosmics.root
  - Other arguments are optional. Which event to plot is an important one.
  - Load it with .L pdspdisp.C (you can compile with .L pspdisp.C++ but it hardly makes a difference).

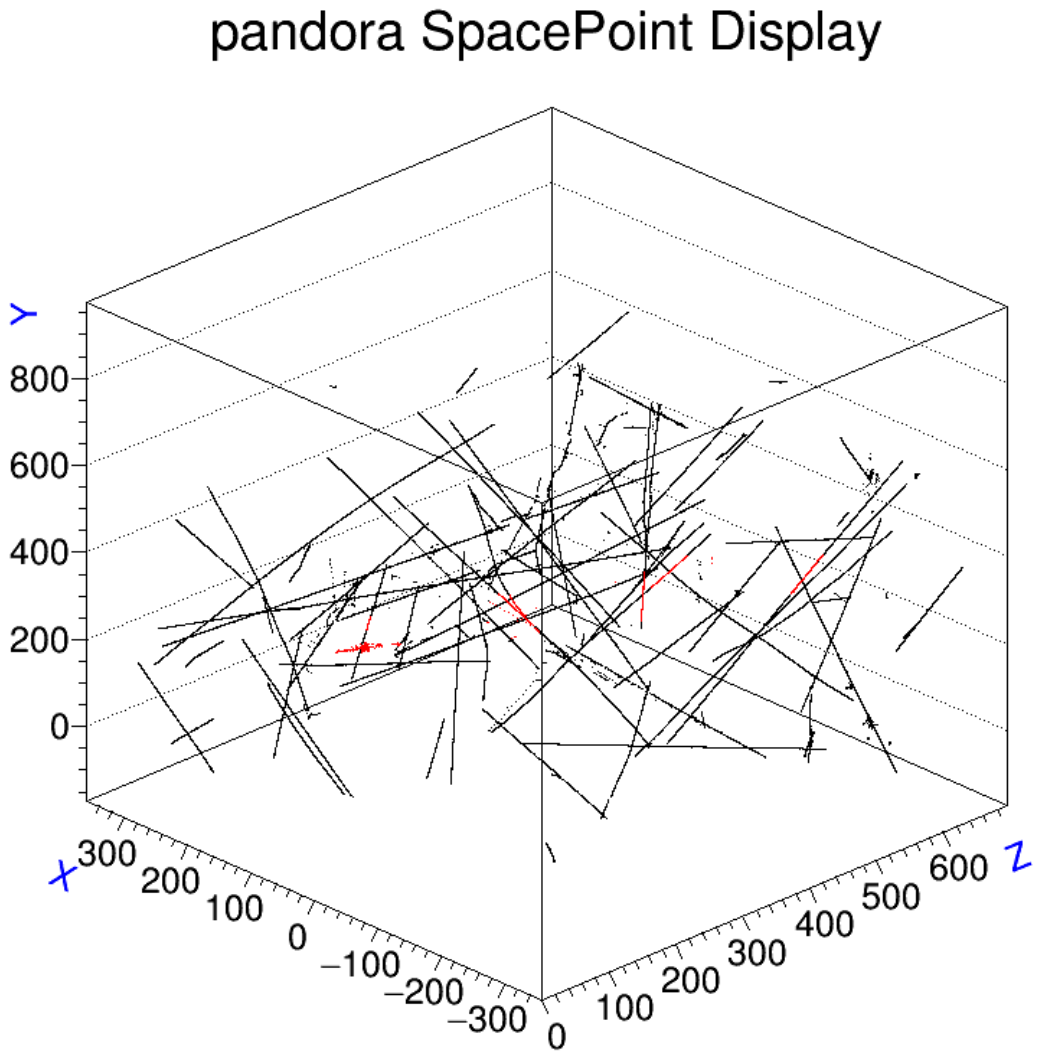
pdspdisp.C output

One argument
toggles pandora
or pmtrack spacepoints

These are TGraph2D's so
you can spin it around
with a mouse.

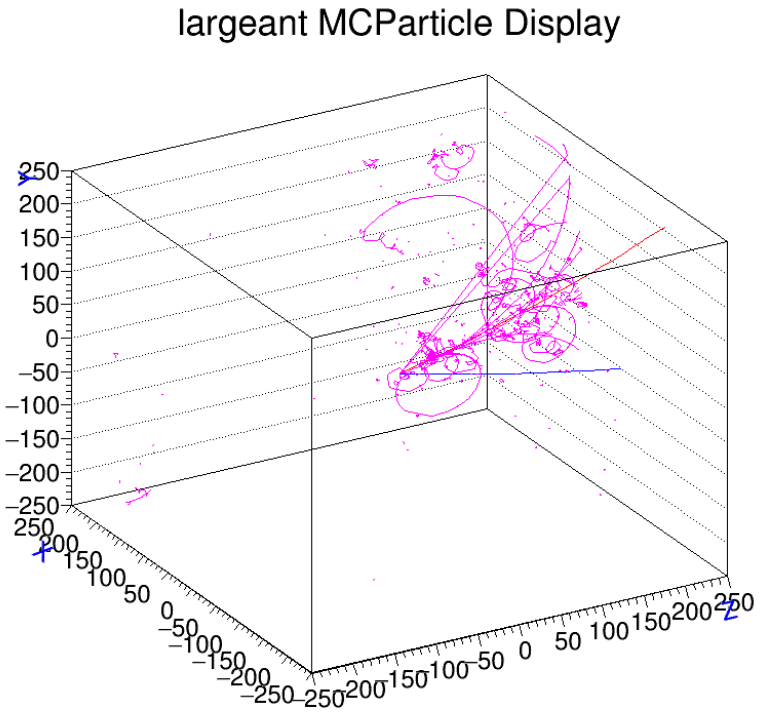
(root leaks memory though
when you do that).

Experiment!



Easy modification to show mcparticle Trajectory points

- I ran some events in gaseous argon by dialing down the density of LAr in a FD workspace job, turned on a magnetic field, and modified pdspdisp.C to visualize the resulting events
- /dune/data/users/trj/gartpc/genielarvsgar/cmc.C is the gallery script

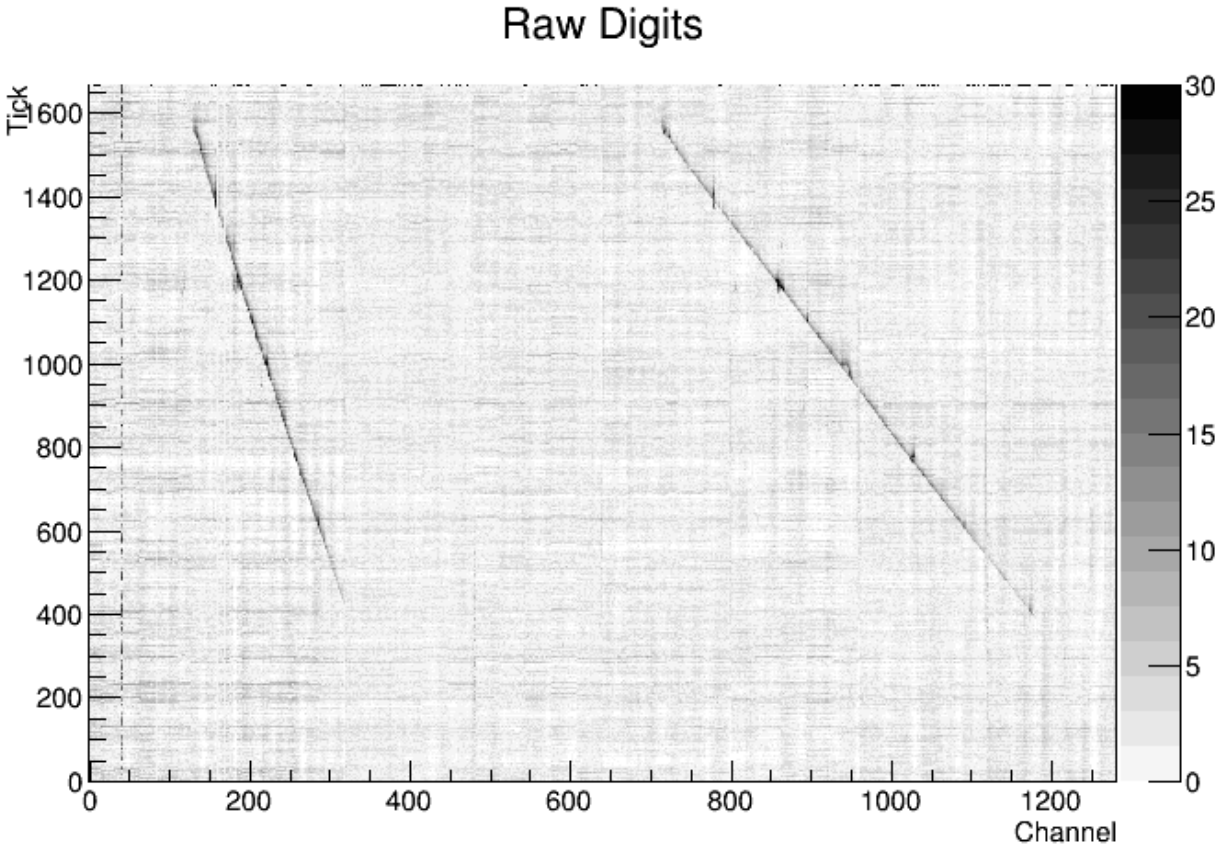


ν_eCC event

magenta: electrons
blue: proton
red: charged pion

Raw Digit Event Display

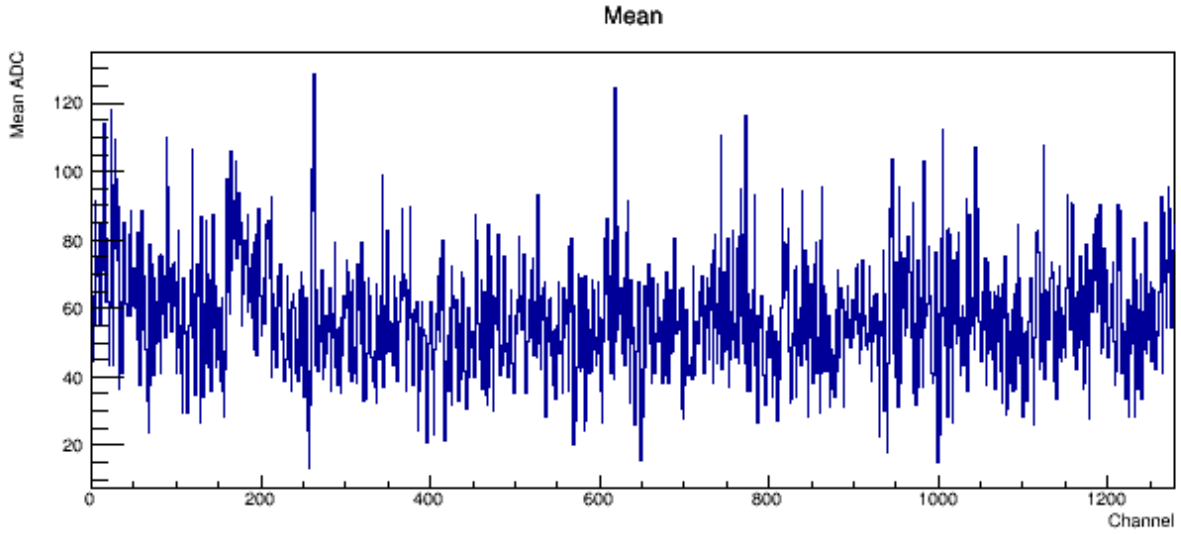
- evd_rawdigits.C
- An imported 3x1x1 data event.
  Look in dune-data.fnal.gov for file locations

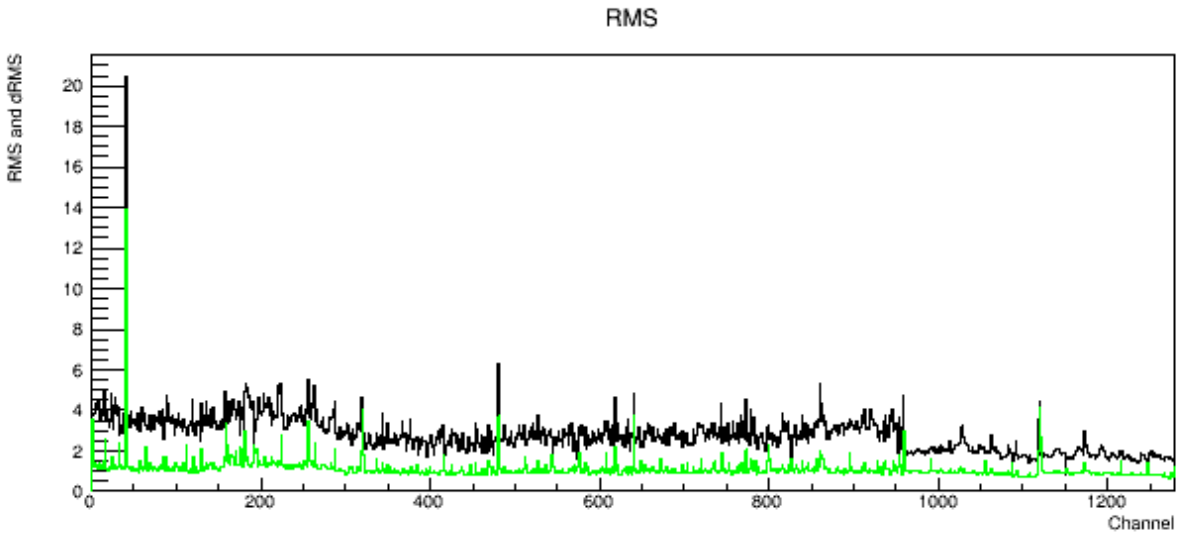


Mean and RMS

meanrms.C

3x1x1 data

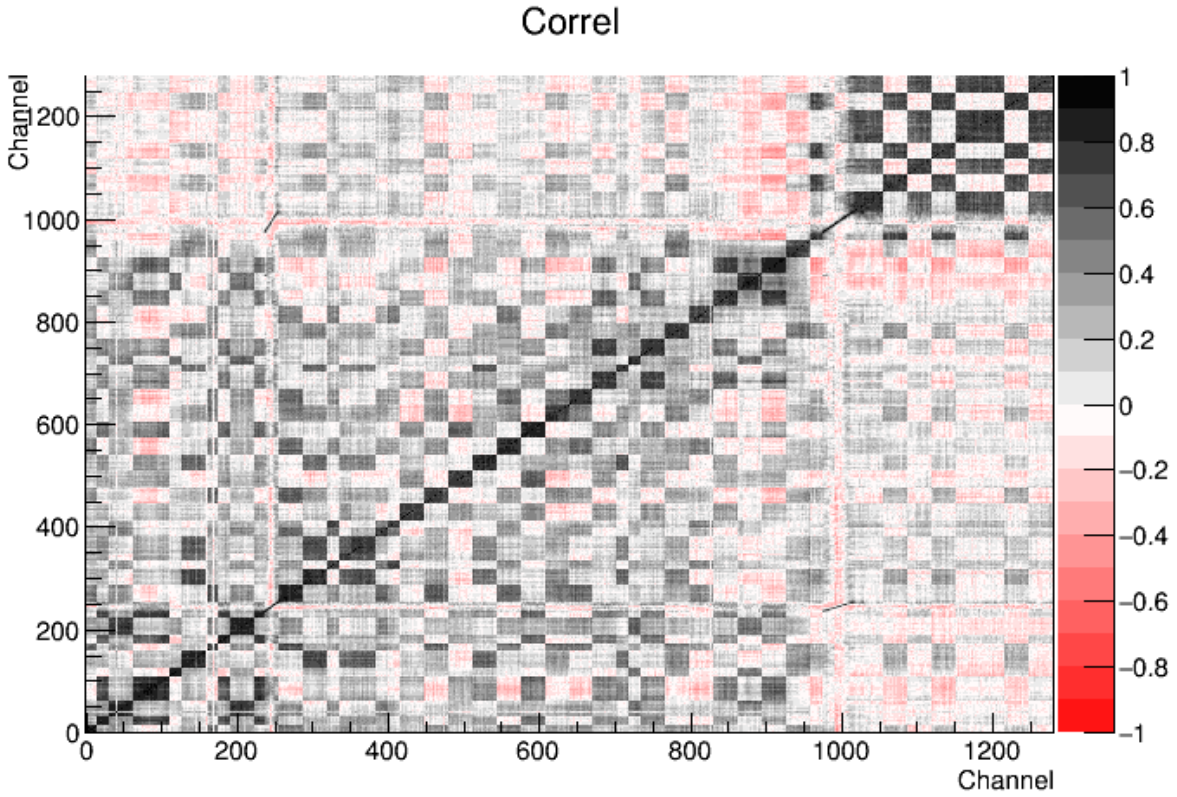




dRMS (green) is the RMS of a channel minus that of its neighbor /sqrt(2)

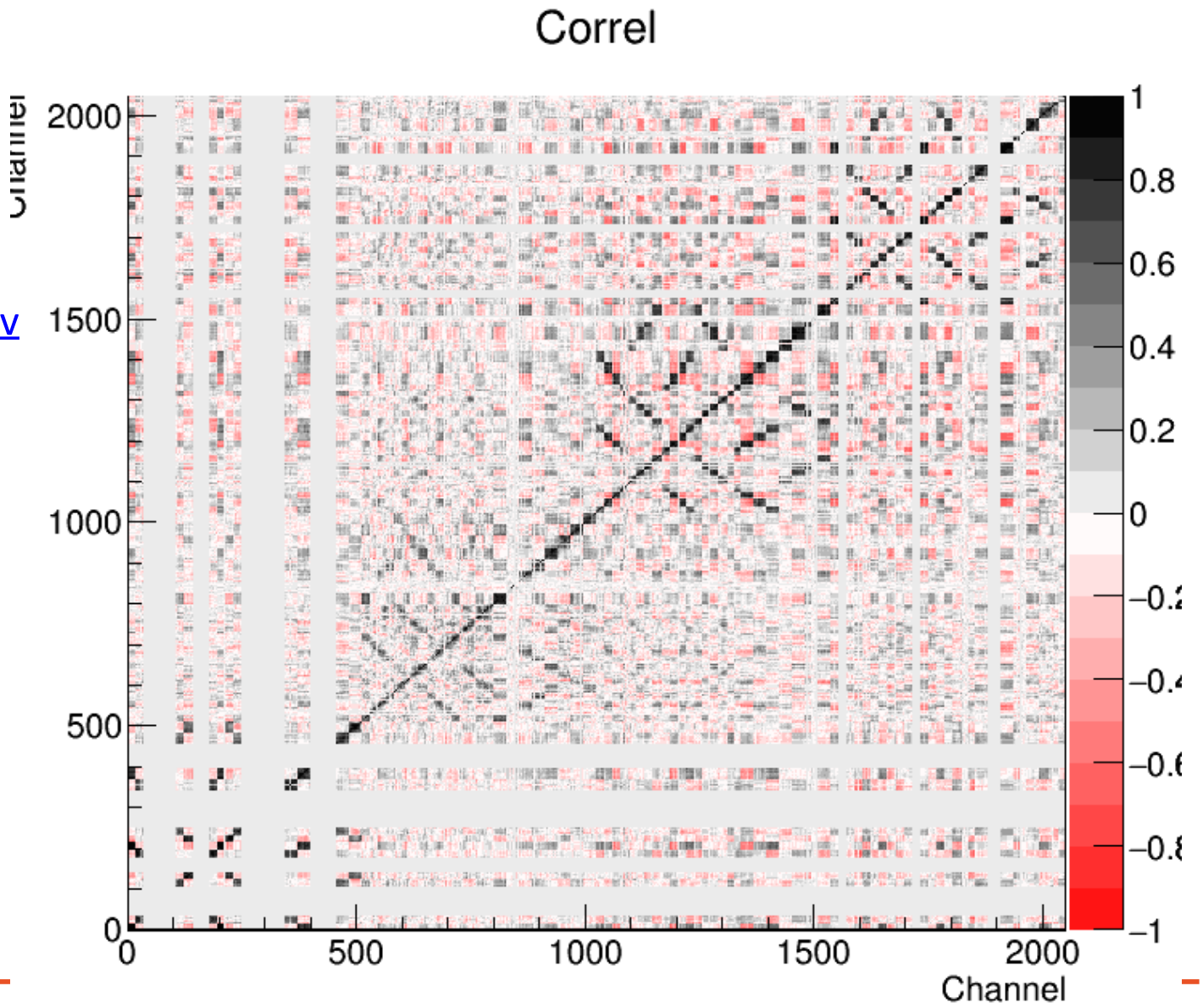
Channel correlations

- chancorrel.C
- Plot of 3x1x1 inter-channel correlation coefficients.
- See dune-data.fnal.gov for tips on how to access 3x1x1 imported data
- 1200x1200 correlations are slow to compute in an interpreted script unless you use ROOT's matrix and vector classes. Then they are super fast!
- Displaying on X Windows is the slow part!



A similar plot for 35-ton

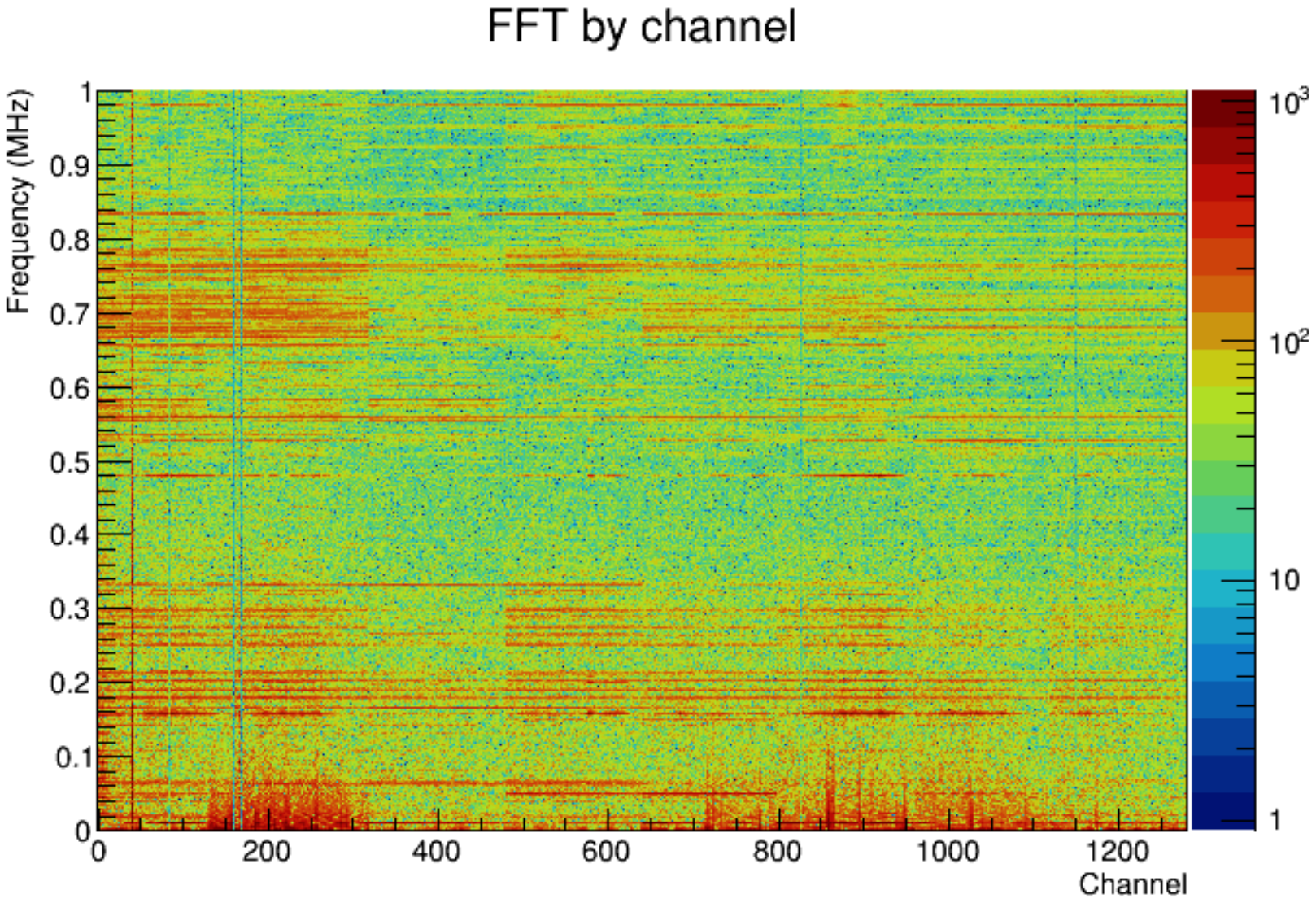
Look in http://dune-data.fnal.gov for examples of how to access 35-ton data. These have been run thruough the splitter input source and thus the channels are "offline" channels



Channel FFT's

chanfft.C

3x1x1 data



Get the Run, Subrun, and Event Numbers in Gallery

Event loop similar to that in Marc's demo.C

```
for (gallery::Event ev(filenames); !ev.atEnd(); ev.next()) {
      auto const& evaux = ev.eventAuxiliary();
      int runno = evaux.run();
      int subrunno = evaux.subRun();
      int eventno = evaux.event();

      std::cout << "Run, subrun, event: " << runno << " " <<
                        subrunno << " " << eventno << std::endl;

  }
```

Try it Yourself

Try it Yourself

- Open up an *art*-formatted root file
- Start a TBrowser
- Look at the Events Tree
- Browse through the branches. They contain the product name, the module label that made them, and the instance name.
- Use the example gallery demo code as a template for retrieving the data members.