How We Get Things Done at SciBooNE

M. Sorel (IFIC – CSIC & U. Valencia)

Workshop on Computing for Neutrino Experiments
FNAL, March 12-13, 2009

SciBooNE

• **Purpose**: precision measurement of muon neutrino-nucleus and antineutrino–nucleus interactions in the ~1 GeV neutrino energy range

• **Timeline:**
- 2005: collaboration formed, proposal
- 2006: detector hall groundbreaking, sub-detectors assembly
- Mar-May 2007: cosmic ray data-taking, detector installation, commissioning
- Jun 2007 – Aug 2008: physics data-taking (neutrinos + antineutrinos)
- Nov 2008: first physics publication

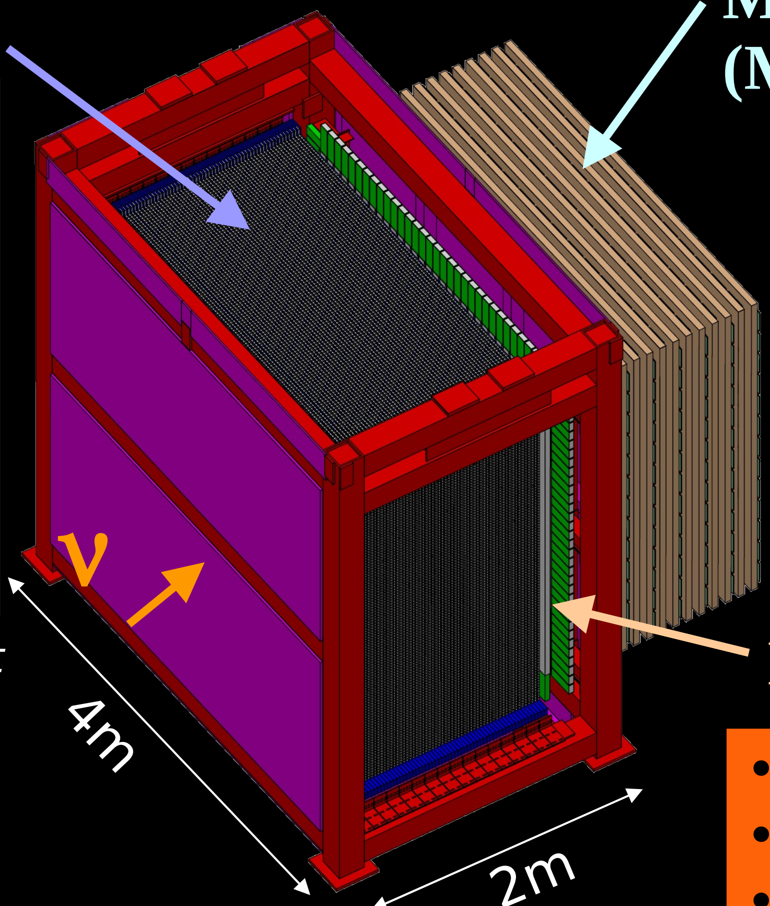
• **People:**
- ~65 collaborators
- ~20 software users, almost 100% of them also software developers!
- All users/developers use computing facilities at Fermilab. Most of those facilities directly maintained by SciBooNE collaborators

SciBooNE detectors

SciBar

- scintillator tracking detector
- 14,336 scintillator bars (15 tons)
- Neutrino target
- detect all charged particles
- p/π separation using dE/dx

Used in K2K experiment



Muon Range Detector (MRD)

- 12 2”-thick steel + scintillator planes
- measure muon momentum with range up to 1.2 GeV/c

Parts recycled from Past experiment

Electron Catcher (EC)

- spaghetti calorimeter
- 2 planes (11 X_0)
- 32 modules per plane
- Identify π^0 and ν_e

Used in CHORUS, HARP and K2K

Framework

- SciBooNE Analysis Framework used for reconstruction
  - This is a fairly heavy-weight SRT-based framework, similar to many other FNAL frameworks
  - Framework core grew out of MiniBooNE framework, but with ROOT-based MIPP/NOvA event data model, and all packages in C++ (no FORTRAN)
- What is used for simulation?
  - Beam: GeANT4 (+ stand-alone FORTRAN code)
  - Neutrino interactions: NEUT AND NUANCE
  - Detector response: GEANT4
  - Digitization: SciBooNE Analysis Framework
- SciBooNE Analysis Framework and/or ROOT macros for data analysis
  - Analyzers left much freedom at this last stage

A full analysis chain from raw data to paper

- A few times per year we reprocess our full neutrino+antineutrino dataset, to provide reconstructed data used by analyzers
- In next slides, real-life example: ~20 framework packages are called, starting from raw data output, to create our default CC inclusive sample
- This follows usual ordering:
  - Reading/merging raw event data
  - Calibrate hits
  - Reconstruct track projections, tracks and clusters within a sub-detector and across 2-3 sub-detectors
  - PID on SciBar tracks
  - Calibration/reconstruction steps interspersed with event filters to cut events uninteresting to all analyses

A full analysis chain from raw data to paper (2)

- **SBDBInterface**: opens connection to MySQL-based offline DB, to extract calibration constants

- **InputDAQFormat**: reads raw detector data (C structs) from the three sub-detectors (SciBar, EC, MRD) and loads it into the framework event

- **MergeAcnet**: reads raw beam data from the “IRM” and “MWR” data streams, calibrates them, merge with detector data if time-stamp matches, check for beam quality & detector alive

- **< EventFilters GoodBeam >:** filter events that do not pass beam/detector quality cuts

- **GeometryDB**: reads geometry info and stores it in framework event, for consistent use in all framework packages (calibration, reconstruction, display, etc.). Not written in event tree. See below about alignment

A full analysis chain from raw data to paper (3)

- **ApplySBCalibration, ApplyECCalibration, ApplyMRDCalibration**: convert hit ADC information into photo-electrons and energy (SciBar, EC, MRD), and hit TDC information into time (SciBar, MRD). Uses calibration constants stored in DB. See calibration later on for more details

- **MRDEventReconstruction**: reconstruct 3D tracks inside MRD

- **sbcat**: reconstruct 2D tracks inside SciBar

- **MatchMRD**: matching individual MRD hits to SciBar 2D tracks

- **Match2DTracks**: reconstruct 3D tracks inside SciBar, from SciBar projections and MRD hits

- **MatchECHit**: matching individual EC hits to SciBar tracks

- **FindECCluster**: construct energy clusters inside two EC planes

A full analysis chain from raw data to paper (4)

•**< EventFilters SciBarTrackMultiplicity >**: require at least 1 SciBar reconstructed track in event for further processing

•**MuonRangeReconstruction**: reconstruct muon range across some/all sub-detectors, and momentum from dE/dx and range-to-energy look-up tables

•**SciBarTrackAttributes**: uses SciBar track dE/dx information for particle identification (MIP-like or proton-like track)

•**<EventFilters MRDMatch>**: keeps interactions originating in SciBar fiducial volume, in-time with beam, and with a muon reaching MRD. This is our CC inclusive sample

•**OutputTree**: write ROOT tree with event info. Can be read back into the framework via a **InputTree** framework package. Data organized in several branches, each one corresponding to a different data processing phase (raw, cal, reco, etc.). Detector-specific folders within each branch

A full analysis chain from raw data to paper (5)

•The above processing is done in three sequential steps, with framework output of current used as input for subsequent

•Event size gets bigger from one processing to subsequent, but fewer events in dataset

•Once beam triggers with no detector activity are filtered (and hits calibrated): ~100 kB/evt, ~100 ms/evt for full reconstruction. MC: larger event size, slower processing

•Some analyses (NC, CC contained in SciBar) obviously do not “final” output, but “intermediate” output

•Definition of analysis-specific samples, data reduction at dataset-level, and extraction (eg, fit) of physics parameters is done in dedicated SciBooNE analysis framework packages and/or in ROOT macros

A full simulation chain from random seed to paper

- Beam MC simulation → event generator → detector MC simulation → SciBooNE Analysis Framework

- **Beam MC simulation:**
  - Use same GEANT4 beam simulation of BNB beamline as MiniBooNE
  - Input from HARP and other hadron production experiments
  - Use stand-alone FORTRAN program to simulate correctly neutrino-producing weak decays, and boost MC statistics
  - Detailed and mature simulation, probably good also for future BNB neutrino experiments!

A full simulation chain from random seed to paper (2)

- **Event generator:**
  - Use both NEUT and NUANCE generator
  - Started with NEUT-only, recently made NUANCE to the same level of maturity for our analyses. Good to have more than one generator for a neutrino scattering experiment!
  - About a year ago, switched from “histogram-based” way of using beam information (flux tables .vs. neutrino flavor, energy, radial position) , to “event-by-event” way. We simulate an interaction for every beam neutrino whose ray-trace crosses the detector, giving it appropriate weight
  - Also need to simulate interactions in dirt material surrounding detector
  - (Partially) account for different nuclear effects in different detector materials: CH, Fe targets

A full simulation chain from random seed to paper (3)

- **Detector MC simulation:**
  - GEANT4-based, written from scratch for SciBooNE. Reads in NEUT/NUANCE-generated events
  - Uses G4 Bertini cascade model for hadronic interactions
  - Very detailed geometry and materials specifications, including alignment
  - SciBar, EC, MRD response:
    - Energy scale (light absolute yield) tuned with cosmic ray data
    - Poisson smearing, single PE PMT resolution accounted for in all sub-detectors charge response
    - Scinitillator quenching, attenuation length in fiber, cross-talk between nearby channels in SciBar charge response
    - Light propagation delays, threshold effects and dead-time in SciBar time response
    - Gain time dependence within beam spill in EC charge response

A full simulation chain from random seed to paper (4)

- Create MC CC inclusive sample in SciBooNE Analysis Framework:

- **InputMonteCarlo**: reads event information produced by detector MC simulation, including MC truth information and detector response. Each event has specific weight from beam, xsec, detector geometry simulation

- Several (5-13) instances of **GetEventWeight**: to re-weight events, either mean event weight, or additional weights for each event, for beam+xsec systematic studies

- **SciBarHitDigitizer, ECHitDigitizer, MRDHitDigitizer**: take hit charge/time response as simulated by detector MC, and convert to ADC/TDC via calibration constants (see calibration later)

- MC now in "raw data output" format, and processing continues as for real data stream

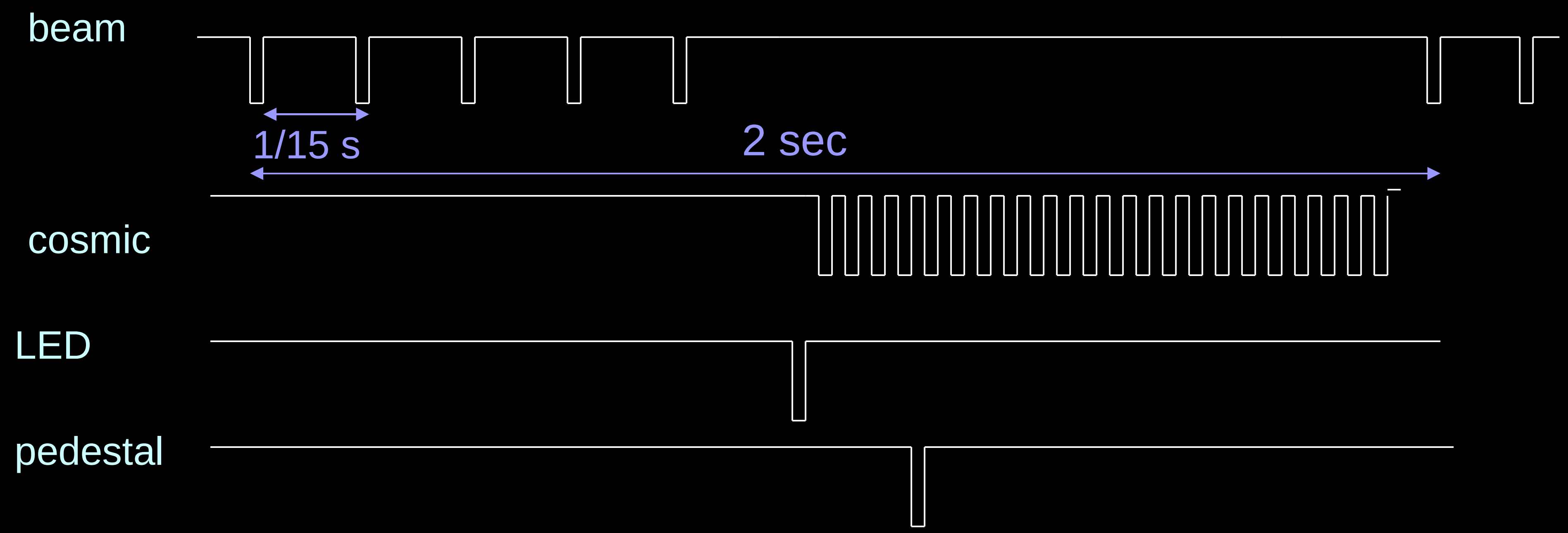
A full simulation chain from random seed to paper (5)

- Only remaining difference with real data stream is that, after reconstruction, the **TrueRecoMatching** package is called: creates links between reconstructed (hits, 2D tracks, 3D tracks, clusters) and MC true objects (hits, particles, showers), sorted with best matches first, for reconstruction performance studies

- Additional MC output through full simulation chain for systematic uncertainties that cannot be evaluated via event reweighting

Calibration procedures

- **Four trigger types**:
  - Beam on target (does not depend on detector activity)
  - Pedestal
  - LED
  - Cosmic ray

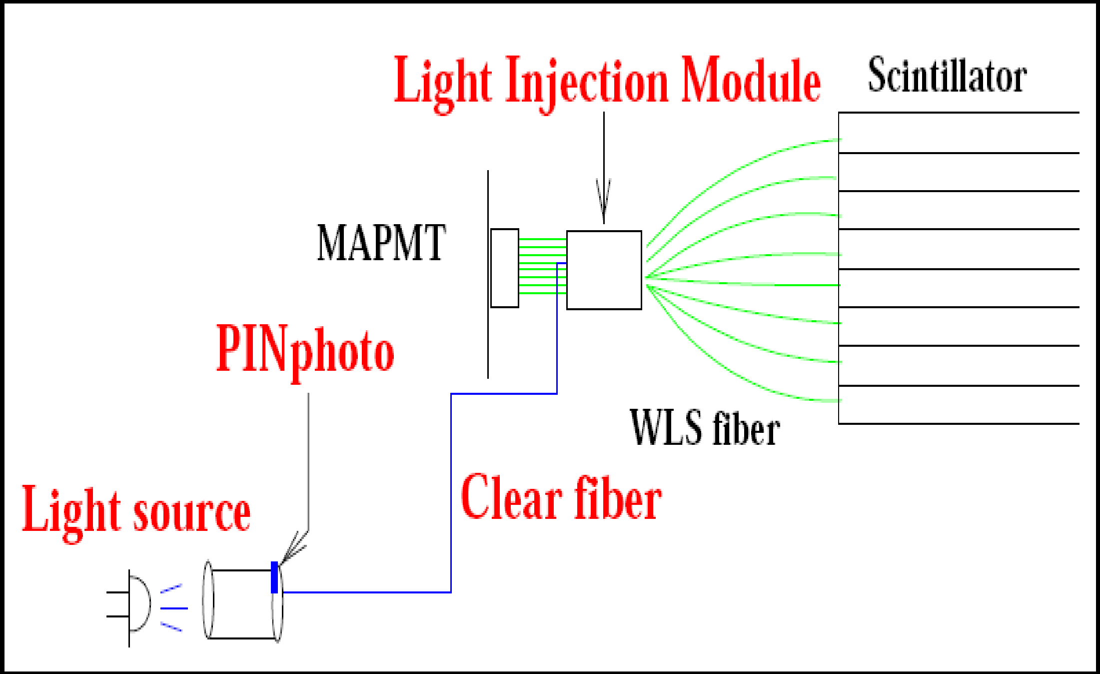


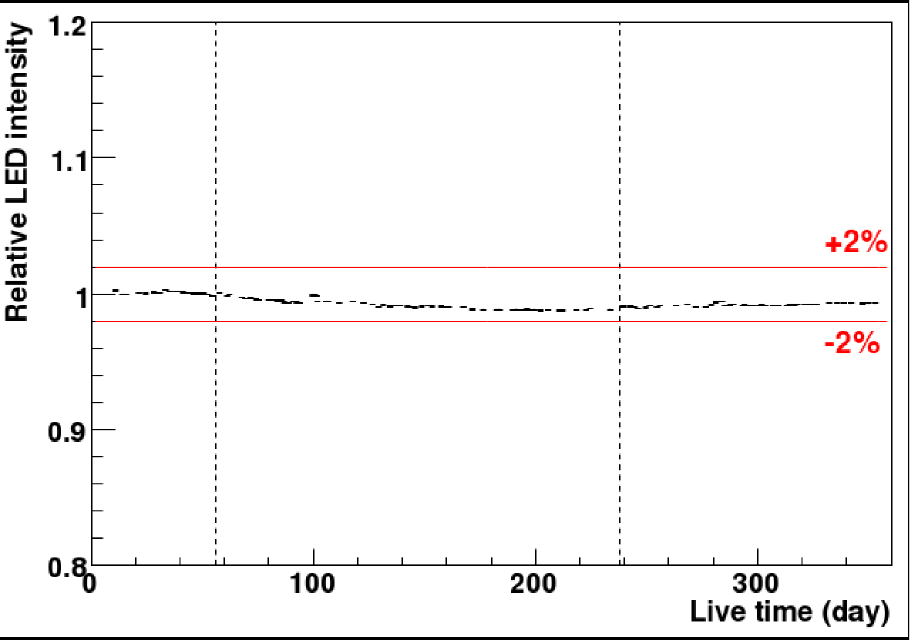
Calibration procedures (2)

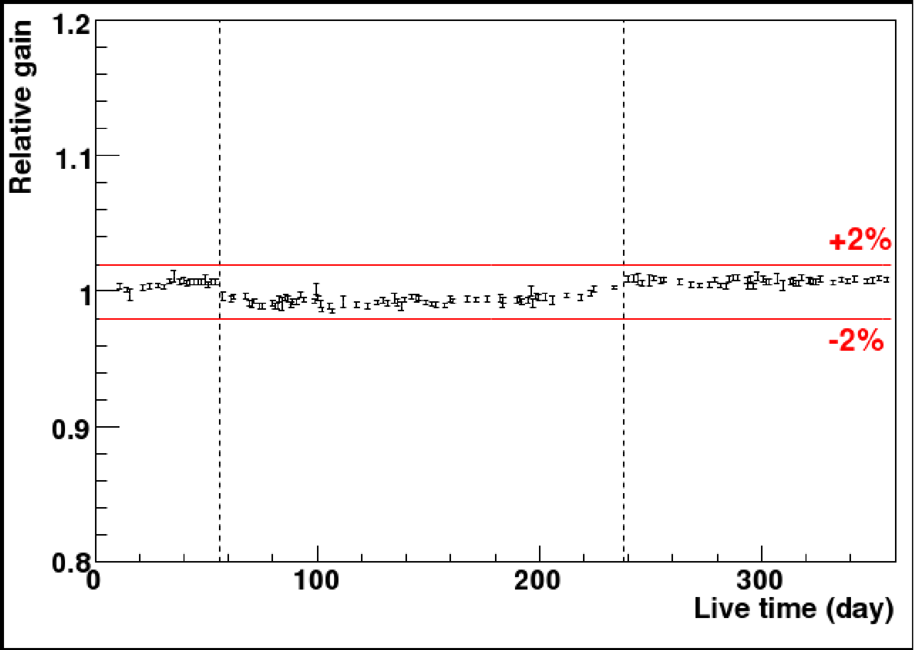
- **Pedestals**:
  - Pedestal trigger after each beam trigger. Compute mean, RMS (or (gaussian sigma) of ADC counts for each channel every 100 pedestal triggers. Store in DB with range of applicability: channel, run, event range
  - In beam triggers, subtract corresponding pedestal value. Proceed with hit calibration if above 3 sigma/RMS of pedestal mean value

- **SciBar ADC to PE:**
  - Single PE ADC distributions for all PMTs prior to installation
  - Monitor and correct for gain drift in time, via a LED-based gain calibration system
  - After each beam trigger, LED trigger fires known amount of light to all channels, allowing to extract the relative gain
  - Relative gain variations are monitored and corrected for every 8 hours with an accuracy of 0.1%
  - Measure cross-talk among nearby channels in MA-PMT in laboratory, correct for it

Details of the gain monitoring system







K. Hiraide, Kyoto U. Ph.D. Thesis (2008)

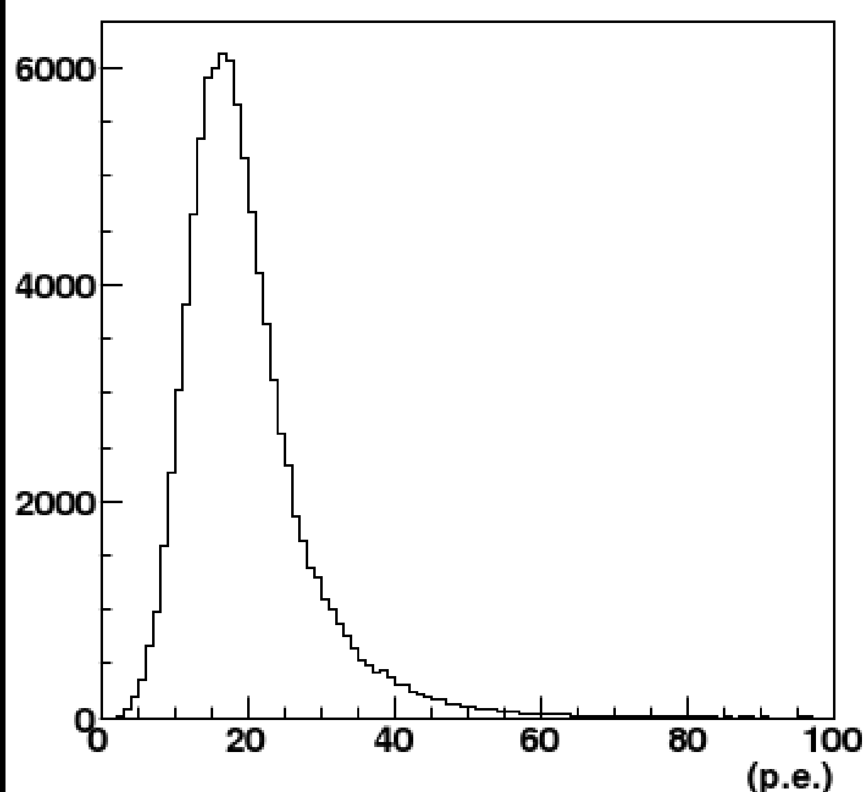
Calibration procedures (3)

- **SciBar overall energy scale (ADC/PE to energy):**
  - Use muons from cosmic ray trigger, also taken after every beam trigger, accounting for attenuation length in fiber, distance crossed within each strip, gain drift
  - Also monitored every 8 hours
  - Light attenuation in fibers measured in K2K, 4% of sample re-measured in SciBooNE, correct for it once position in strip is known (after reconstruction)

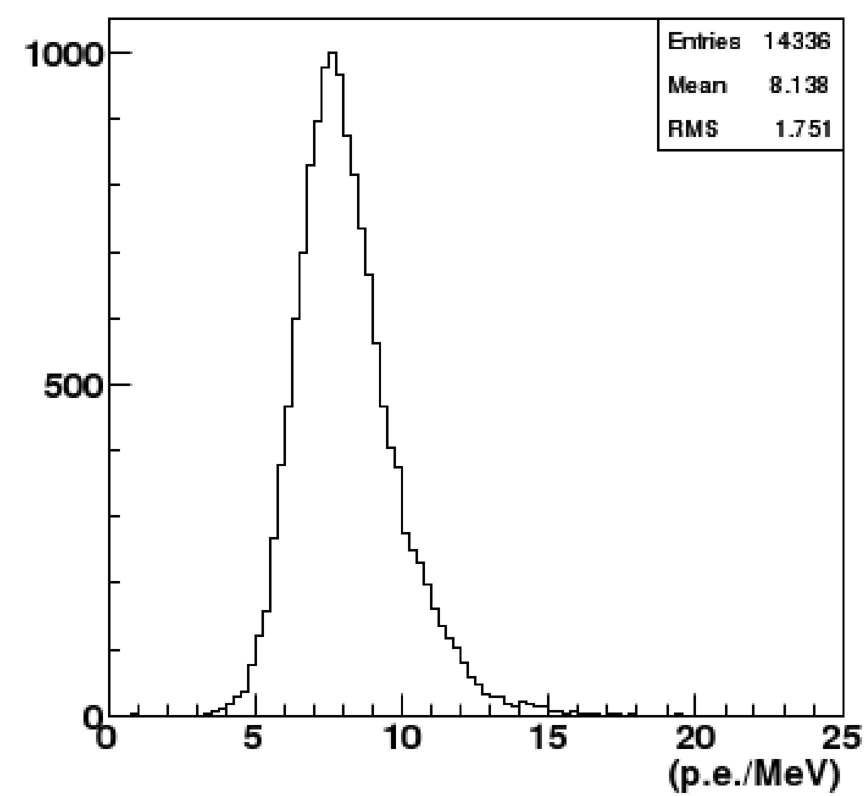
- **SciBar TDC to time:**
  Use cosmic ray muons to extract QT correction (time-slewing) and t0 for all channels

Details of cosmic ray calibration

Number of PEs for cosmic ray muons for a typical channel



Energy calibration constants for all channels



K. Hiraide, Kyoto U. Ph.D. Thesis (2008)

Calibration procedures (4)

- **EC**:
  - Cosmic ray muons to correct for different PMT gains
  - Neutrino interactions in SciBar and reaching EC/MRD, to extract gain dependence on beam bunch number (due to different integration of signal overshoot)
  - Signal read at both fiber ends to eliminate dependence of light attenuation

- **MRD**:
  - Channels' t0 for TDC to time conversion with cosmic muons
  - Also have MRD ADC information, but we don't use it much for now

- Calibration constants above stored in DB, typically channel-by-channel, several of them keeping track of time dependence

Snapshot of DB web interface

phpMyAdmin

Database

sbdb (53)

sbdb (53)

- eccal
- eccal1
- eccal1_bkp
- ec_adc_ped
- ec_adc_ped_bkp
- ec_event_rate
- k2k_sbalig
- k2k_sbalig_global
- k2k_sbamt2
- k2k_sbcal
- k2k_sbcal_adc2e
- k2k_sbcal_att
- k2k_sbcal_ener
- k2k_sbcal_index
- k2k_sbcal_pe
- k2k_sbcal_runindex
- k2k_sbcal_t0
- k2k_sbgatetiming
- k2k_sbtqcal
- k2k_sbtqcal_indx
- k2k_sbvata
- mrd_adc_cal
- mrd_adc_ped
- mrd_adc_ped_bkp
- mrd_isci2elec_map
- mrd_isci2elec_map_bkp
- mrd_tdc_cal
- mrd_tdc_cal_bkp
- mrd_tdc_offset
- mu_range_to_ke_iron
- mu_range_to_ke_polystyrene

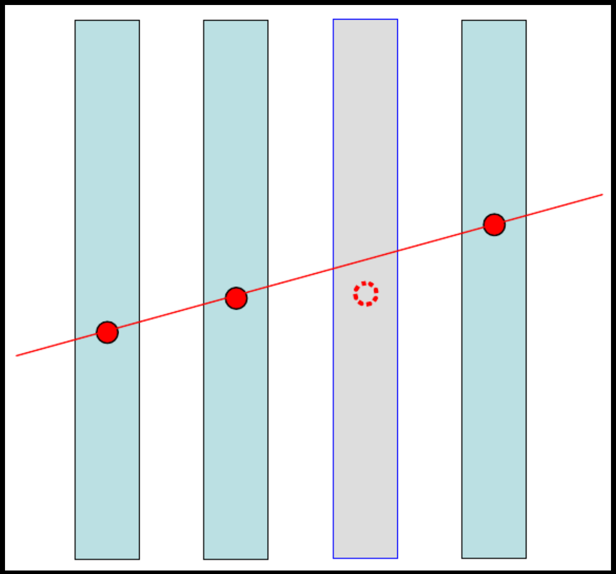
mu_range_to_ke_polystyrene	145	MyISAM	8.7 KiB	-
pot_per_simulatedevent	1	MyISAM	2.1 KiB	-
pot_summary	2,106	MyISAM	74.4 KiB	-
sbcal	157,696	MyISAM	4.7 MiB	-
sbcal_att	14,336	MyISAM	127.0 KiB	-
sbcal_bkp	14,336	MyISAM	183.0 KiB	-
sbgeom_ecalignment	256	MyISAM	10.5 KiB	-
sbgeom_ecmodule	23	MyISAM	4.0 KiB	-
sbgeom_GlobalPos	4	MyISAM	2.4 KiB	-
sbgeom_mrdmodule	25	MyISAM	4.2 KiB	-
sbgeom_scibaralignment	64	MyISAM	3.8 KiB	-
sbgeom_scibarmodule	9	MyISAM	2.8 KiB	-
sbvata	14,592	MyISAM	243.3 KiB	-
scibar_adc2mip	14,336	MyISAM	243.0 KiB	-
scibar_adc2pe	14,336	MyISAM	243.0 KiB	-
scibar_adc_ped	242,190,547	MyISAM	9.5 GiB	-
scibar_event_rate	1,389	MyISAM	44.5 KiB	-
scibar_gain_monitor	20,772,208	MyISAM	854.0 MiB	-
scibar_led_monitor	5,852	MyISAM	303.6 KiB	-
scibar_tdc_offset	5,430	MyISAM	199.6 KiB	-
test_table	14,336	MyISAM	295.0 KiB	-
test_table_index	0	MyISAM	1.0 KiB	-
week_def	55	MyISAM	2.0 KiB	19 B
53 table(s) / **Sum**	**316,652,496**	**MYISAM**	**11.7 GiB**	**280.2 KiB**

Alignment procedures

•Relying on survey data solely for EC and MRD (~1 mm accuracy), taken in assembly hall at CDF, and later in detector hall with detector installed

•For SciBar, checked survey data with alignment based on single cosmic ray tracks

•Single layers (out of 64) masked one-by-one, residuals in x and y between measured and track-extrapolated positions computed for each one. Iterative procedure

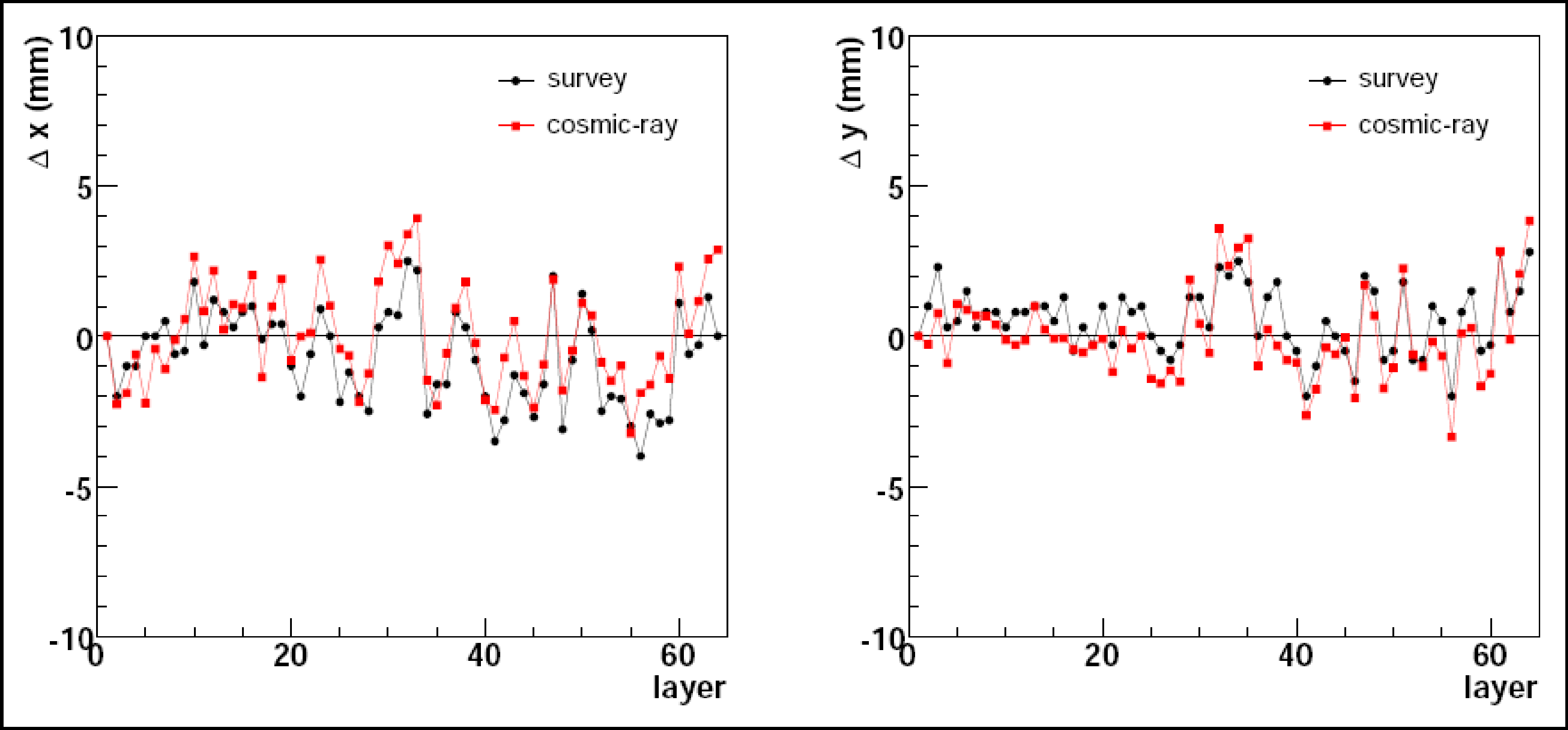


•Agrees well with survey, within accuracy of method

•Nominal values for layer z positions, rotations, strip positions within layers

Alignment procedures (2)

•SciBar layer transverse position alignment: comparison between survey data and cosmic ray data



Credit: H. Takei, Tokyo Tech. U.

FORTRAN or C++?

- We use almost exclusively C++ code in SciBooNE offline

- Only FORTRAN “exceptions” to this I am aware of:

- Monte Carlo:
  - Neutrino parent re-decay routines in beam MC simulation
  - NUANCE generator

- SciBooNE analysis framework:
  - Library to read K2K events, which were in modified ZEBRA (ZBS) format. Useful to exercise framework code in early SciBooNE days
  - NUANCEInterface code, calling NUANCE routines to perform neutrino xsec reweighting
  - Linked also with CERNLIB, but in practice we use GSL or ROOT for numerical libraries

What worked really well?

- SciBooNE Analysis Framework:
  - Modularity and structure made collaborative effort of developers very successful. Also, code never “grew out of control”
  - Regular freezing of framework releases, allowing all analyzers to be “on the same page” at all times
  - Version control not only for code, but also for job configuration files
  - Event data model
  - Database interface

- Simulation:
  - Re-use existing code we like where available (eg, beam simulation)
  - Deal with two event generators (NEUT and NUANCE)
  - Modifications to read single beam events in generators
  - Decision not to adapt old G3 detector MC code from K2K, but re-write in G4, offering easier hadronic physics tuning

What did not work so well?

•Interaction with FNAL Computing Division:
sometimes decided to pursue “home-grown, physicist” solutions rather than more professional CD solutions in the interest of having something working quickly. We hope to change this (eg, TB server)

•“Heavy-weight” framework code, “old” framework core:
installation at remote institutions/laptops possible, but painful. Also, parts of code developed ~2000 and start to show signs of time

•Documentation/tutorials incomplete or out-of-date:
extra burden on experts to train newcomers, discouraged some senior people to get involved in analysis

•Fixed-format event data summaries existing, but not always working and useful as they should be

Backups on Computing Infrastructure

Data Size/Rates, MC Needs

- Total data size and rate:
  - ~5 TB worth of raw data, accumulated over full data-taking period
  - About half in beam triggers, half in calibration triggers
- MC needs:
  - ~10 times data statistics for default MC (interactions in detector)
  - Additional MC to simulate dirt interactions
  - Additional MC to simulate a few (~5) detector response systematic variations

Central FNAL Systems

- CPUs used:
  - ~80 modern CPUs, in ~20 desktop or rack-mounted machines at WH10X
  - Rack-mounted machines only for Condor batch jobs
  - Desktop machines for interactive use of Condor batch jobs
  - Scientific Linux Fermi LTS release 4.4/4.5, 32-bit architecture machines

- Storage used:
  - 2 rack-mounted RAID systems currently in use, with 7.4 TB capacity each, currently in WH10X and administered by us
  - Working toward moving this system, together with new rack-mounted RAID system of 10 TB capacity, to Feynman Computing Center
  - Few TB scratch/analysis space on desktop machines at WH10X
  - Only accessible on FNAL site
  - Raw data also on dCache/Enstore

- CPUs and storage used for all three tasks:
  - Reconstruction and data filtering
  - MC production
  - User analysis