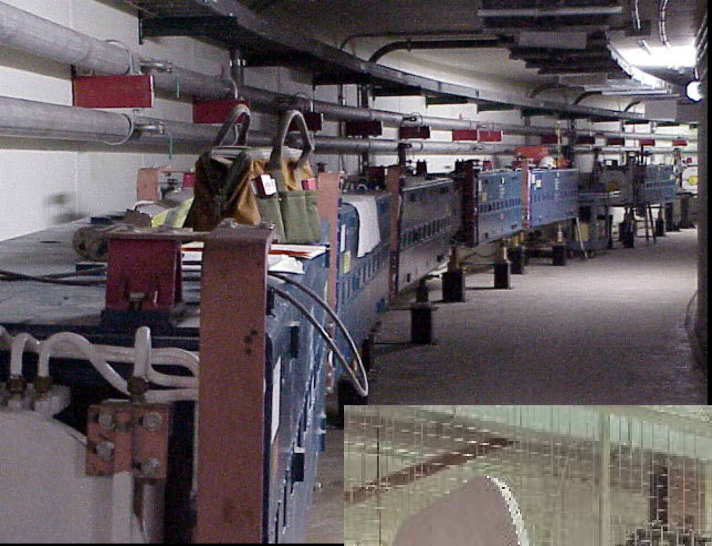
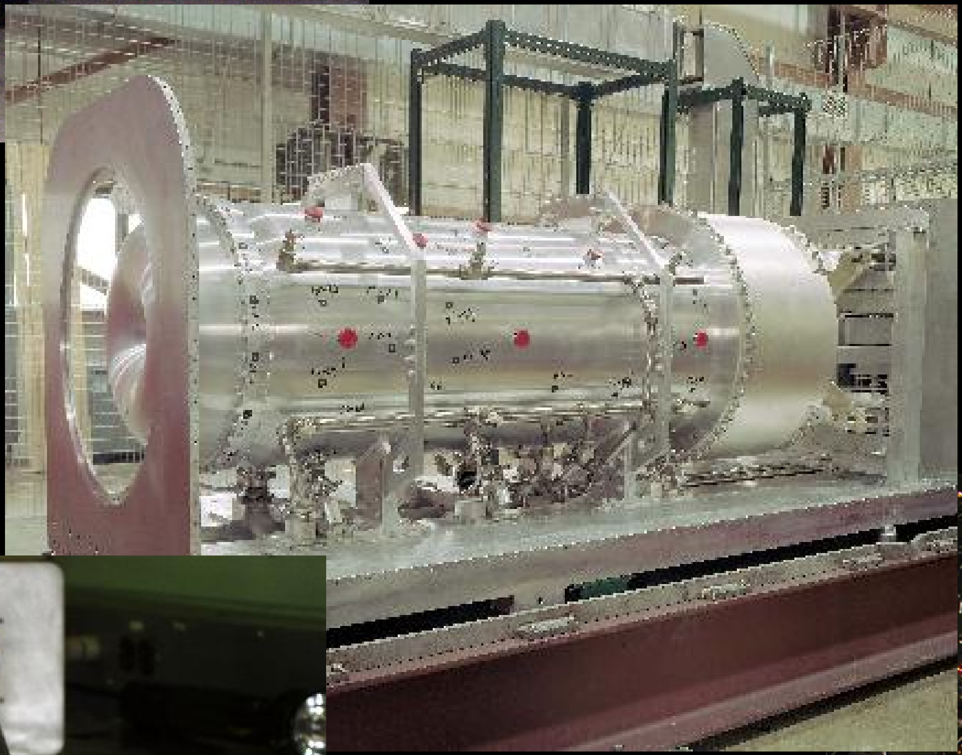
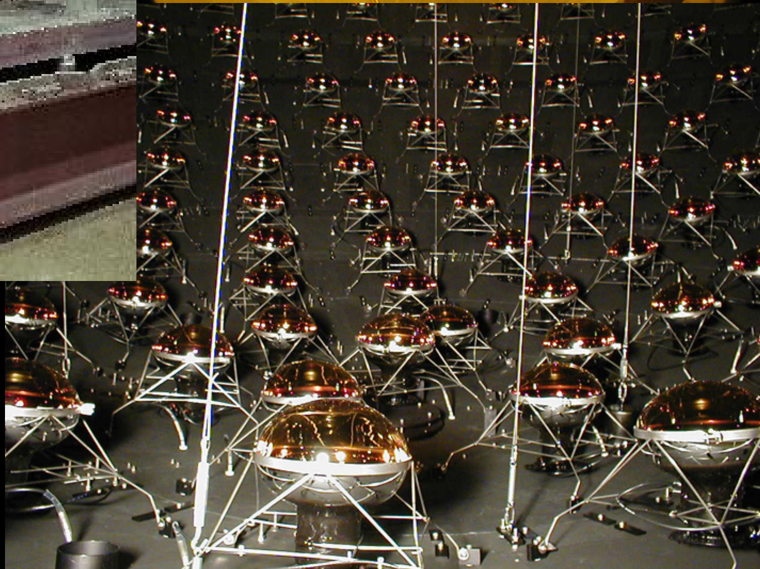
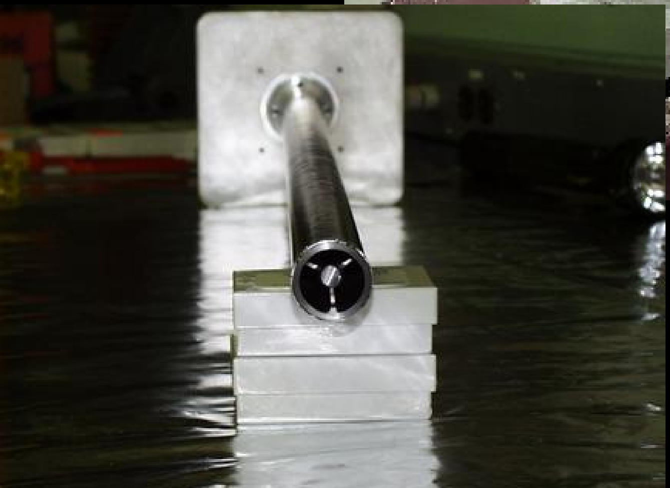
MiniBooNE Computing & Analysis Model

Chris Polly, Univ. of Illinois









MiniBooNE Purpose and Timeline

- Purpose
  - Oscillations: general search for ν_e appearing in a ν_μ beam over short baselines
  - Cross sections: high statistics ν measurements around 1 GeV
- Timeline
  - Dec 1997: Proposal submitted
  - Oct 1999: Construction starts
  - Aug 2002: First beam delivered
  - Jan 2006: 5.6E20 POT collected (after data quality cuts), switch to anti-nu beam
  - Jul 2007: First oscillation results published
  - Jul 2007-present: 8 more publications
  - Future:
    - 3 papers nearing completion (NuMI events in MB, anti-nu appearance, $CC\pi^+/CCQE$ ratio)
    - many more papers ~4-6 planned this year (especially in xsec realm)
    - official anti-nu running ends at June shutdown
      - 6.6e20 collected w/ nu mode, ~5e20 collected w/ anti-nu beam
    - proposal to run another 5e20 anti-nu beam (2-3 yrs at current rates)

MiniBooNE Collaboration

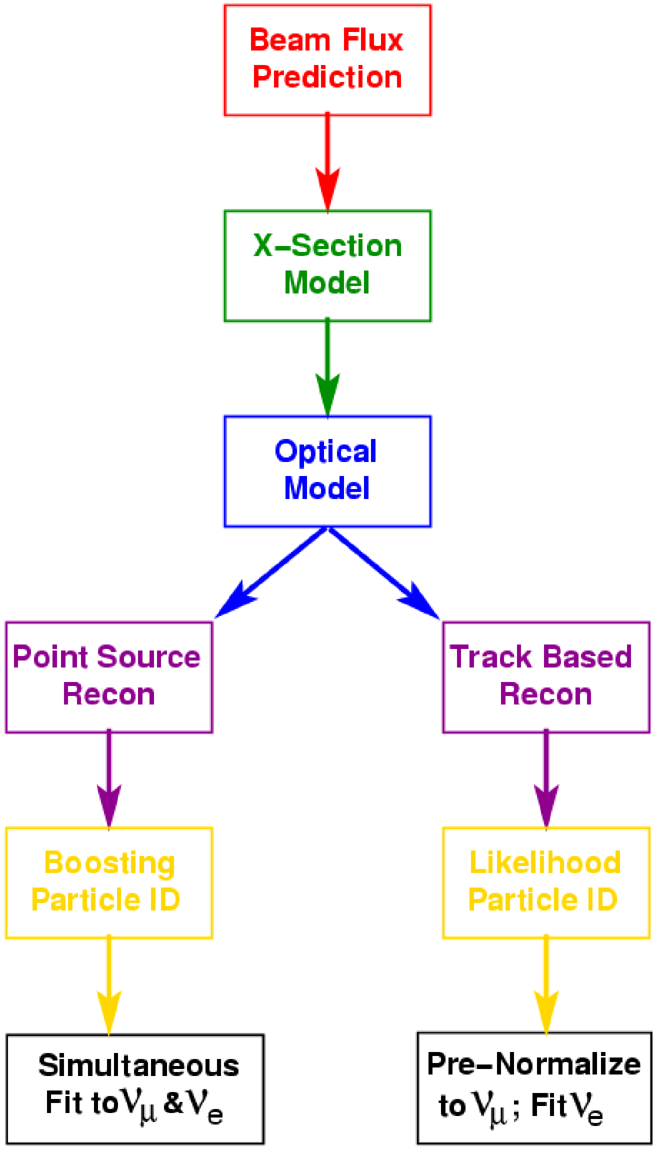
A. A. Aguilar-Arevalo[e], C. E. Anderson[p], L. M. Bartoszek[g],
A. O. Bazarko[m], S. J. Brice[g], B. C. Brown[g], L. Bugel[e],
J. Cao[ℓ], L. Coney[e], J. M. Conrad[e], D. C. Cox[i], A. Curioni[p],
Z. Djurcic[e], D. A. Finley[g], B. T. Fleming[p], R. Ford[g],
F. G. Garcia[g], G. T. Garvey[j], C. Green[j,g], J. A. Green[i,j],
T. L. Hart[d], E. Hawker[j,c], R. Imlay[k], R. A. Johnson[c],
G. Karagiorgi[e], P. Kasper[g], T. Katori[i], T. Kobilarcik[g],
I. Kourbanis[g], S. Koutsoliotas[b], E. M. Laird[m], S. K. Linden[p],
J. M. Link[o], Y. Liu[ℓ], Y. Liu[a], W. C. Louis[j], K. B. M. Mahn[e],
W. Marsh[g], P. S. Martin[g], G. McGregor[j], W. Metcalf[k],
H.-O. Meyer[i], P. D. Meyers[m], F. Mills[g], G. B. Mills[j],
J. Monroe[e], C. D. Moore[g], R. H. Nelson[d], V. T. Nguyen[e],
P. Nienaber[n], J. A. Nowak[k], S. Ouedraogo[k], R. B. Patterson[m],
D. Perevalov[a], C. C. Polly[i], E. Prebys[g], J. L. Raaf[c], H. Ray[j,h],
B. P. Roe[ℓ], A. D. Russell[g], V. Sandberg[j], W. Sands[m],
R. Schirato[j], G. Schofield[k], D. Schmitz[e], M. H. Shaevitz[e],
F. C. Shoemaker[m], D. Smith[f], M. Soderberg[p], M. Sorel[e 1],
P. Spentzouris[g], I. Stancu[a], R. J. Stefanski[g], M. Sung[k],
H. A. Tanaka[m], R. Tayloe[i], M. Tzanov[d], R. Van de Water[j],
M. O. Wascko[k 2], D. H. White[j], M. J. Wilking[d], H. J. Yang[ℓ],
G. P. Zeller[e,j], E. D. Zimmerman[d]

~80 physicists from ~18 institutions

- Analysis and simulation mainly performed on FNAL clusters via local and remote login:
  - User-based analysis run on MB-maintained ~100-node Condor cluster
  - Mass reprocessing and MC simulation performed via FNAL GP Farm priority running (400 node) and opportunistic cycles on FNAL-based OSG (D0, CDF, CMS)
  - Some clockticks used at Colorado, Princeton, and LANL
  - More details on computing infrastructure in Chris Green's talk

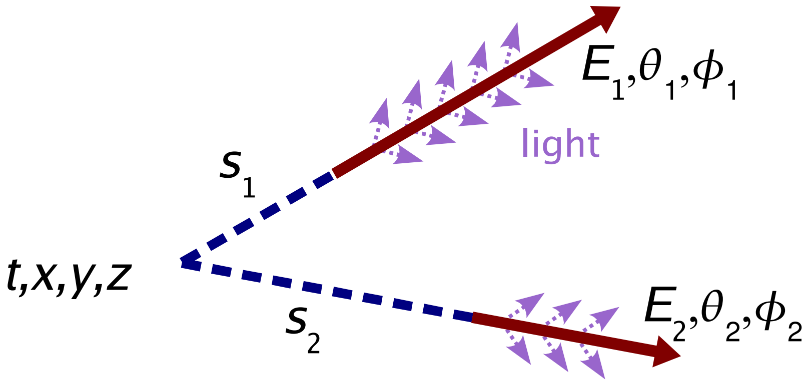
MiniBooNE Analysis Chain & Software

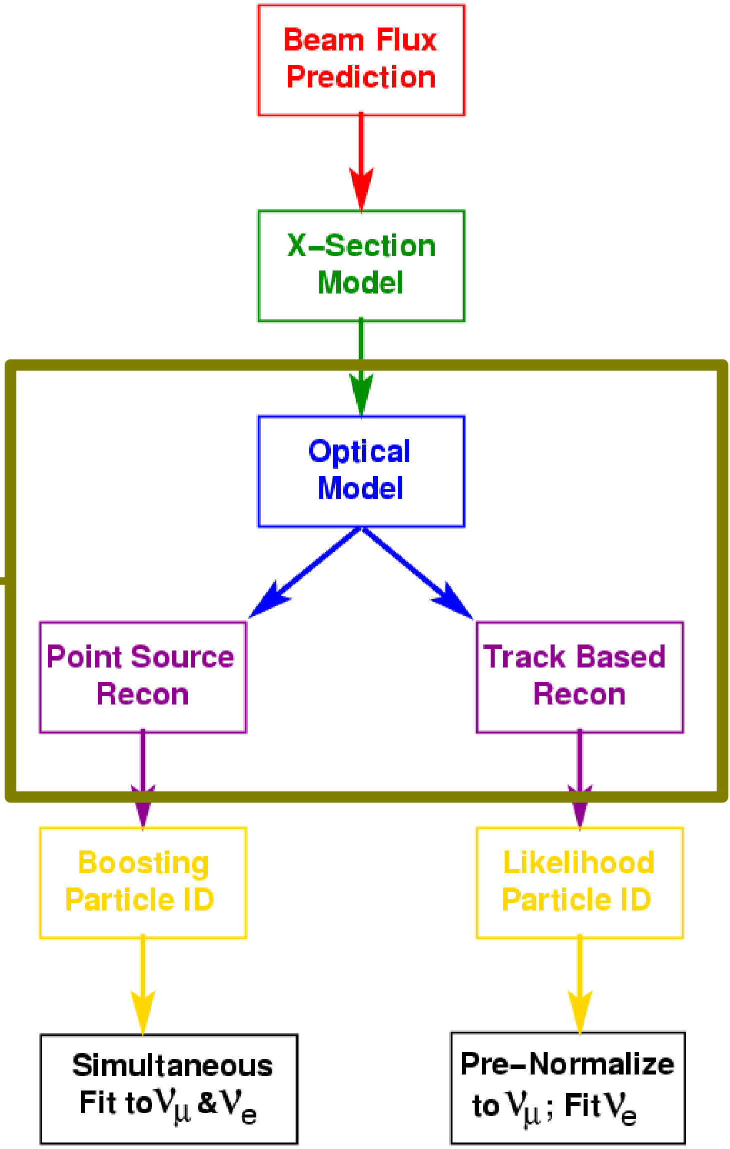
- ✔ Geant 4: Used to produce π and K produced at the target (and elsewhere), and subsequent ν flux
- ✔ Nuance: Predicts rates and kinematics of various NC and CC ν interactions
- ✔ Geant 3: Tracks final state particles delivered by Nuance, models light propagation in the tank
- ✔ AF: Several quasi-independent reconstruction algorithms (point and track-based). Data and MC are both made to enter analysis chain here
- ✔ AF: Several method of particle ID developed (likelihood and boosted decision tree based.
- ✔ AF: Systematic err analysis and final fits to E_ν performed. At this point, work becomes user-based and mainly performed in desktop Condor



Where we spend our CPU cycles

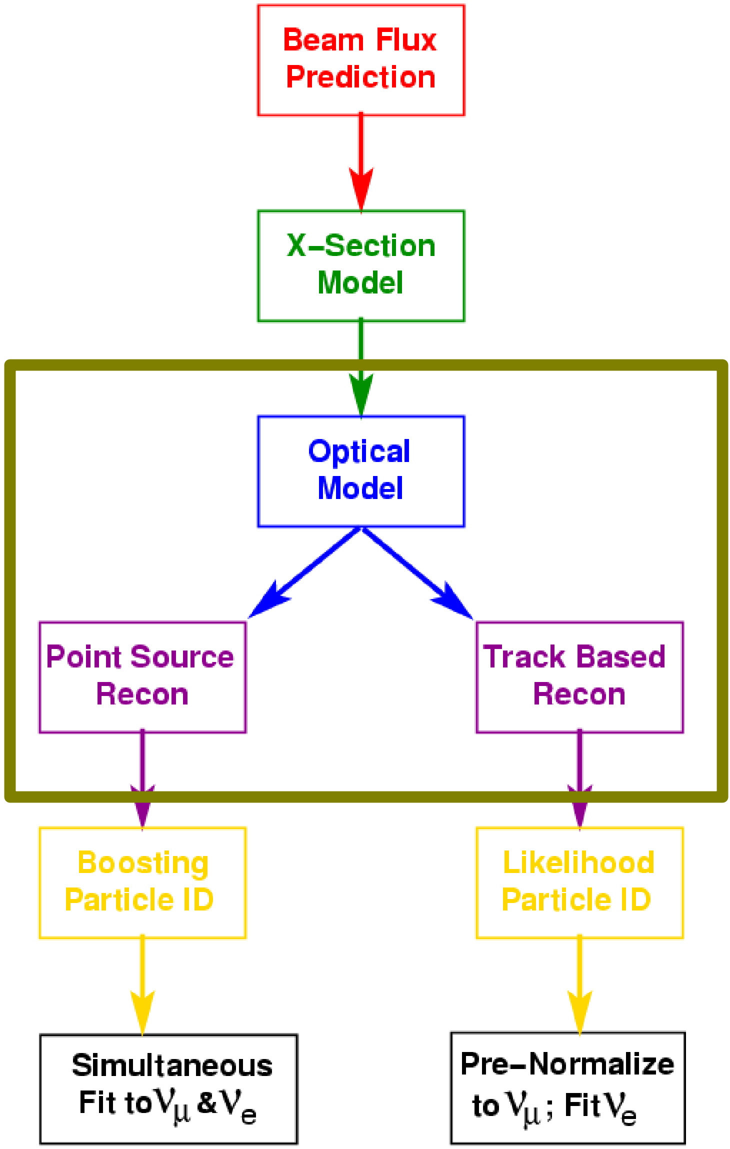
- Vast majority (95%) of CPU time spent here:
  - ~20% on tracking particles and propagating light
  - ~80% on reconstruction
    - About half of reconstruction time spent on reconstructing NC π^0 events





Where we spend our CPU cycles

- MC uses orders of magnitude more CPU than data
  - Simple reason: bulk of time is spent in recon... needed for both MC and data
  - Typical MC baseline
    - Run several (2-3) central value MC simulations for each data reprocessing...each 10-20 x data stats
    - Run several (4-6) unisims, single parameter variations, to assess systematic effects...each 10-20 x data stats
    - Run ~100 multisim worlds, multi-parameter variations, to assess systematic errors...each run at 1-2 x data stats
  - Add them al up and MC total is ~160-380 times the number of data events
  - Also means storage needs dominated by MC



The MiniBooNE Analysis Framework

- Analysis Framwork credits
  - Originally developed by Jim Kowalkowski and Marc Paterno
  - Help in implementing ROOT from Phillipe Canal
  - Deployed for MiniBooNE by Chris Green, lot's of scriptwork
  - Maintained by Steve Brice
  - Same initial AF structure as CDF, D0, and SciBooNE with some subsequent divergence
  - Analysis packages written by MiniBooNE users
- Some of the key Analysis Framework features
  - Analyzer code available to all through CVS repository
  - Many 'frozen' release of tagged AF versions
    - 71 major versions since start of MB, just as many 'sub'versions, i.e. 71-0 and 71-1
  - Behavior of packages controlled via RCPs
    - Every package has a standardconfig
    - Every package has many other pre-defined configs (other common run modes)
    - Users can always create a myconfig to flip code switches/parameters as needed
  - Standard output to a ROOT tree
    - Also options to run AF and send output to simple ROOT or PAW ntuples
    - Very important feature for giving all users a comfortable analysis environment

The MiniBooNE Analysis Framework

- Another key analysis feature is the modularity
  - Trivial for users to check out any package (with some scriptwork probably from Chris Green)
    - Can make changes to version in home area
    - Recompile picking up local packages ahead of standard releases
    - Run a myAnalysisFramework in place of standard Analysis Framework
  - Also trivial to create a new package (thanks again to some minor scriptwork)
    - Any package has (read) access to the output from any other package
  - Trivial for any new package to add a 'chunk' of data to the standard output stream
- All package code, rcps, and output chunk contents CVS browsable

Typical analysis chain...MC specific

- Start with flux prediction from Geant 4 beamline/horn MC with pion production tables taken from HARP, and kaon production taken from world data
  - Output takes the form of four flux histograms for nu and anti-nu (e and mu)
- Redecay program...the odds of a π or K making a ν that passes through MB are low. Would take a long time to track every particle. Instead when a meson is made that produces a neutrino that passes through MiniBooNE, the last stage where the meson decays is redecayed 1000 times.
- Nuance randomly samples the flux histograms, randomly picks a neutrino cross-section (subject to relative probabilities of course),
  - Provides a list of all final state particles leaving the struck nucleus
- Final states are handed to G3 to track and model light propagation
- BooNEglob program...because the original G4 parent info is not passed through Nuance a giant look-up table is used to match the specific neutrino energy to a list of possible parents from G4. If done randomly the effect is the same as having just passed the info.
- The MC info from Nuance, BooDetMC (G3), and BooNEglob is all encoded into ntuples that are read into the Analysis Framework
  - The framework then runs an MCthroughDAQ package to turn MC info into faux versions of UncalibratedData (quads)
  - That data is then turned into CalibratedData and overlayed with strobe triggers

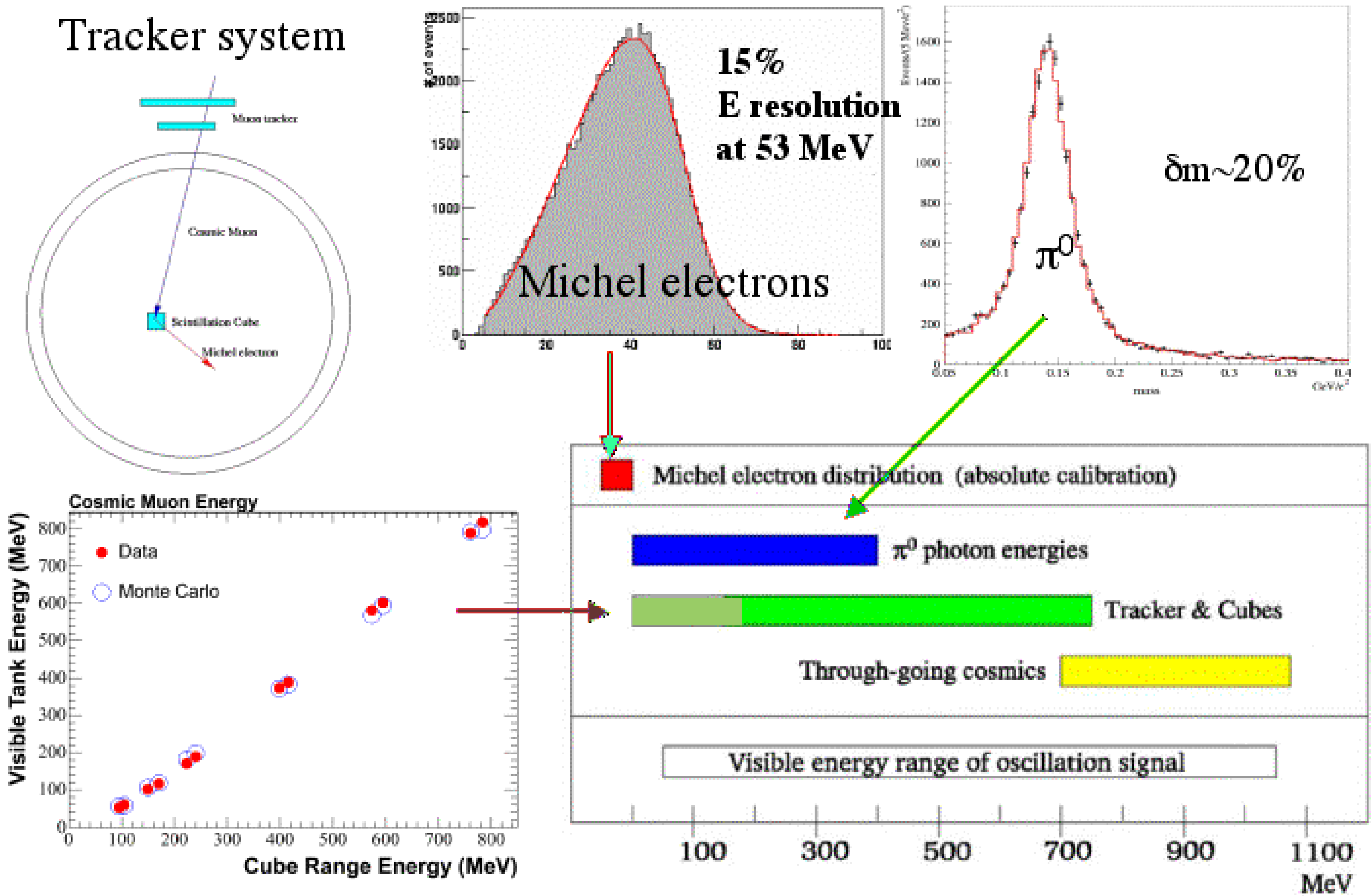
Typical analysis chain...data specific

- Raw data is read in from the DAQ output, includes event-by-event ACNET info
- Connection established to PostGresQL database
  - Calibration constants, run quality info from ACNET and logs, etc.
- Calibrate beam diagnostic info from toroids, resistive wall monitor, BPMs
- Apply data quality cuts and track stats of DQ cuts
  - Beam position, horn performance, log info, etc.
- Run uncalibrated through calibration procedure
  - Calibration information for every PMT maintained through continuous 3.33 Hz flashing of laser
  - Laser intensity kept to level where <10% of tubes have non-zero pe, ensures calibration data is ~single pe

From the CalibratedData stage on, data and MC events are treated identically!

Many other calibration sources besides the laser to get an understanding of our ability to model events...

Calibration Sources



FORTRAN or C++

- Mixed
  - G4 beam MC in C++, but G3 detector MC in FORTRAN
  - Nuance written in FORTRAN
  - All user-developed AF modules and data chunks required to be in FORTRAN, but core AF code is C++
  - Standard output to ROOT tree, in principle can be analyzed directly via ROOT macro...although painfully
  - More common to parse standard data reprocessing and MC file with output going to ROOT or PAW ntuple and then use ROOT or PAW macros

What worked well and what didn't?

- Modularity and ease of contributing to the Analysis Framework was great
- Being able to provide PAW or ROOT output ntuples (in addition to standard ROOT tree) critical for keeping everyone involved in the analysis
- FORTRAN requirement for AF packages
  - Benefit of making interactions between packages and chunks...making maintenance easier
  - Most postdocs and grad students not a big fan of working in FORTRAN...lot's of homegrown ways to do post-production analysis
- PostGresQL database running on one machine incapable of keeping up with several hundred jobs running on the OSG and trying to connect back
  - Developed offline database that could be shipped to each worker node
  - Worked but better to have an improved DB
- G4 Beam, Nuance, G3 Detector
  - Nice in the sense that it allowed each group to work in an enviroment they were comfortable with
  - Had to handle handshaking at input->output stages
  - Some less than ideal solutions like BooNEglob