CLAS12 Software Status

Veronique Ziegler

CLAS12 Collaboration Meeting
Jefferson Lab
November 2, 2016

Overview

- Current Status
  - software overview
  - simulation
  - event reconstruction
  - event visualization & analysis tools
- Timelines and Milestones
- Summary

Computing Model and Architecture

- ClaRA (CLAS12 Reconstruction & Analysis framework)
  - Multi-threaded analysis framework based on a Service Oriented Architecture upgraded to use x-Msg → ClaRA-4.3 (c.f. Gagik's talk at the workshop)
  - Reconstruction compliant with ClaRA-4.3
  - Service-based physics applications composition
  - Verified linearity of ClaRA scaling up to physical core limit with the number of threads (next slides)
- Reconstruction code framework and running environment
  - Detector reconstruction and event building framework → plugins chained into a reconstruction application
  - Reconstruction services deployed in ClaRA platform
  - coatjava-3.0 runs ClaRA platform from command line – reconstruction can be run in multi-threaded mode

CLAS12 Software at a Glance Reminder

I/O package
- Evio I/O provider
- Raw event decoder
- ccdb access tools

Plotting package
- Histogramming & Fitting
- Ntuple Maker
- Event viewing & monitoring tools

Geometry package
- Geometry Objects and Methods
- Detector Geometry

Reconstruction packages
- CVT: central tracker
- DC: hit-based & time-based trk^g
- HTCC: e- ID
- FTOF: timing
- EC/PCAL: e- & neutrals ID
- FT-Cal, -Hodo: low angle e-, neutrals
- EB: detectors track matching & PID

Analysis package
- Kinematic Fitter
- Event Selector
- Fiducial Cuts provider
- Fast MC

Simulation (GEMC)
- Event simulation emulating detector responses and track propagation through CLAS12

Event viewer
- cLAS eVEnt dISPLAY

Utilities
- DC noise finder
- MagField & Swimmer

Clas
Offline
Analysis
Tools

GEMC 2.5 Release

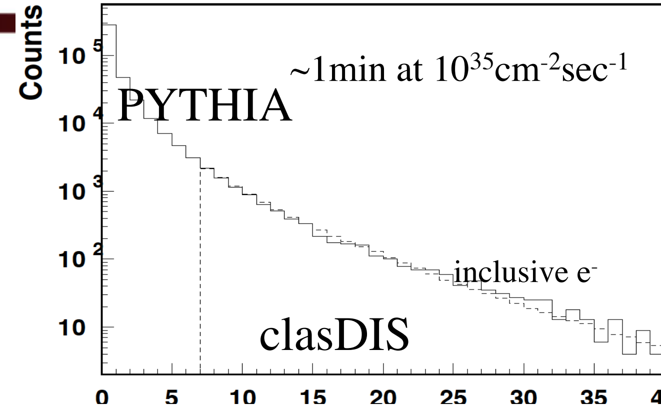
M. Ungaro

- FADC Output
- Output organized by CRATE/SLOT/CHANNEL
  - use translation table (idem data) for decoding to SECTOR/LAYER/COMPONENT/TDC, ADC
- Geometry from coatjava geometry package, CAD import capability
- Advanced digitization routines
  - EC/PCAL: CCDB calibration constants, attenuation length, effective velocity, voltage, strip volumes
  - FTOF: CCDB calibration constants, Attenuation length, effective velocity, time walk, smeared / unsmeared output, status
  - DC: CCDB calibration constants, Intrinsic inefficiency, time resolution, drift velocity
  - SVT: energy sharing between strips, 3-bit ADC
  - CTOF: CCDB calibration constants, attenuation length, effective velocity, time walk, smeared / unsmeared output, gain balance, geometry input from CAD
  - HTCC: Mirrors reflectivity from actual measurements, PMT q.e. function
  - LTCC: Mirrors reflectivity from actual measurements, PMT q.e. function, C4F10 Refractive index, transparency
  - MM: updated geomtry variation, Lorentz angle from solenoid actual value. Energy sharing
  - FT: Beam line components from CAD, Calorimeter, Hodoscope, MM Tracker, advanced digitization
- Further details about downloads, gcards, targets, see gemc.jlab.org and Mauri's talk at the workshop:
  https://www.jlab.org/indico/event/180/

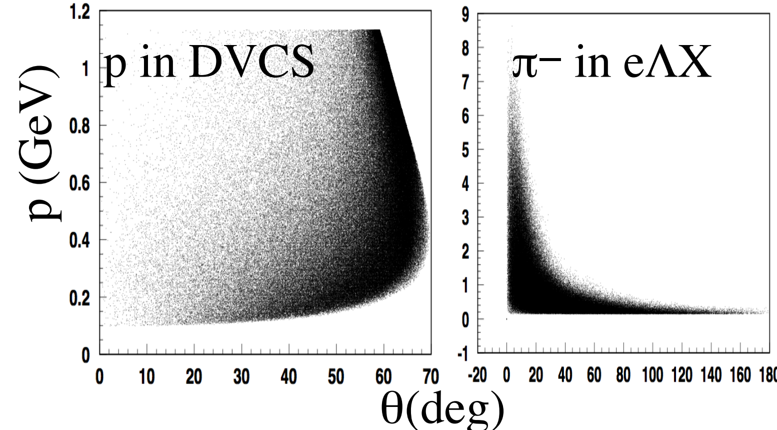
CLAS12 Event Generators

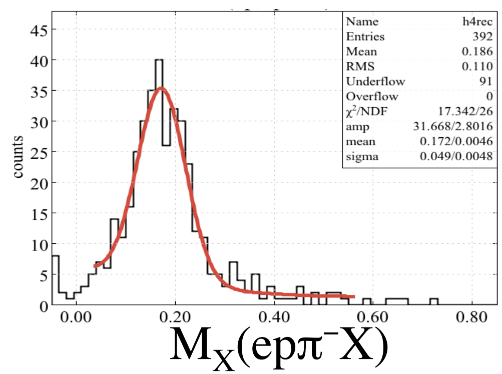
H. Avakian

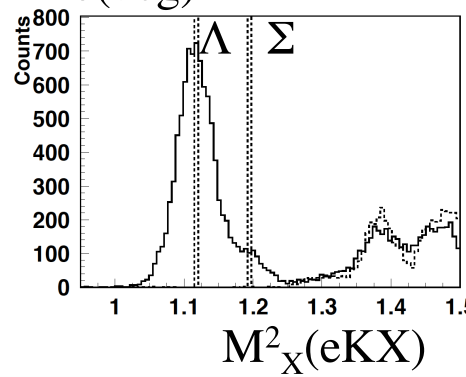
Generated events available for calibration and reconstruction software tests /group/clas12/mcdata/generated/lund/:



- SIDIS LUND MC (PYTHIA and PEPSI)
  - Generating (claspyth) low Q2 events for hadronic background and PID studies using modified PYTHIA
  - Generate (clasDIS) single and double-spin dependent processes using the modified PEPSI (LEPTO)
- Exclusive events
  - exclusive γ (DVCS), π/η using GPD models
  - exclusive eKΛ, epππ,..







Reconstruction Status

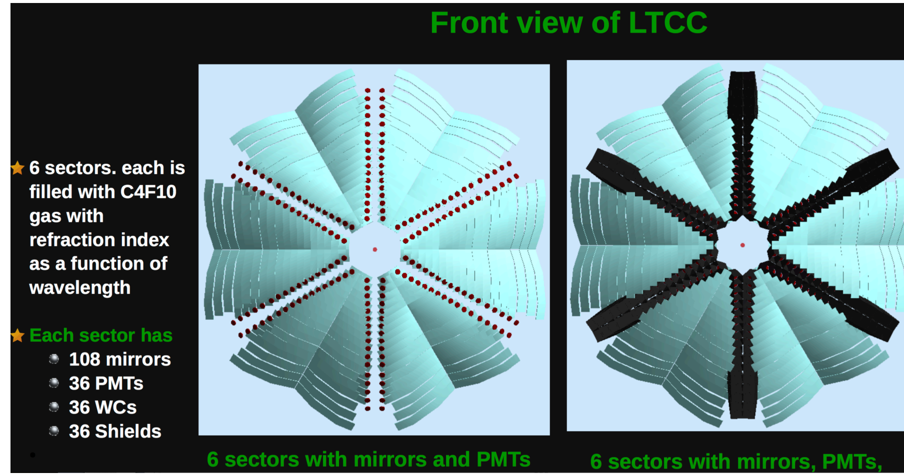
- Detector constants initialization from ccdb & run-dependent calibration constants loading.
- Runs reconstruction services as plugins. Available services for:
  - Central Tracking
    - CVT reconstruction with new geometry from code developed by Peter Davis (U. Surrey); alignment with Millepede on cosmics (J.Gilfoyle).
  - Forward Tracking
    - DC reconstruction → realistic time smearing. Vertex reconstructed at distance of closest approach to the beam line. Improved Hit-Based Tracking (segment finding and track fitting).
  - TOF
    - Combined package for CTOF and FTOF. Cluster-matching algorithms for FTOF 1a and 1b. Geometry implementation from geometry package (A. Kim (UConn))
  - EC/PCAL
    - Validated geometry. Peak and cluster energy splitting algorithm in place (C. Smith, G. Gavalian). Geometry implementation from geometry package (A. Kim (UConn))
  - HTCC
    - Validated GEMC mirror reflectance (N. Markov (UConn)).
  - Forward Tagger
    - Calorimeter and Hodoscope reconstruction available (R. deVita (INFN)).
  - Event Builder
    - Matching of tracks in the drift chambers with the detectors in the forward detector
    - Identification of trigger particle. Uses trigger particle vertex time with RF time to get event start time. (J. Newton (ODU))

LTCC Development

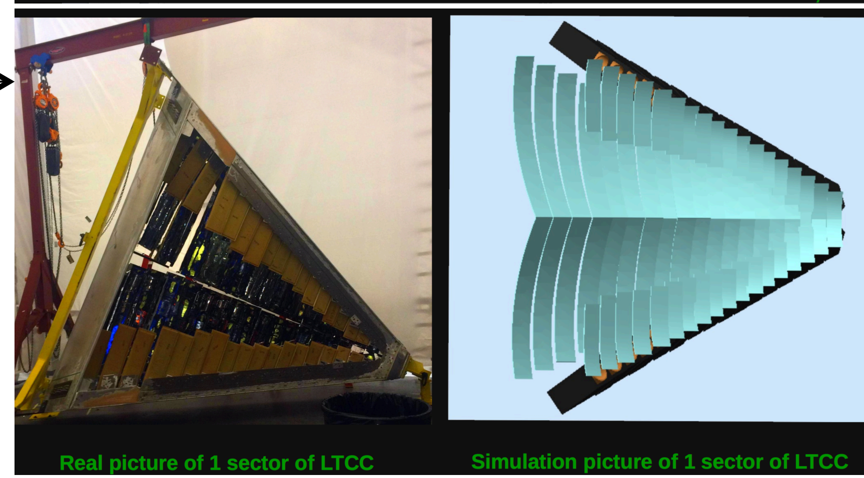
- Presentation at CLAS12 Workshop by Burcu Duran (Temple U.)

LTCC Simulation Status

- Construction and placement of Winston Cones ✔
- PMT Shields ✔
- LTCC Box (material in active areas) in progress
- Geometry ✔



- Digitization and response function in progress



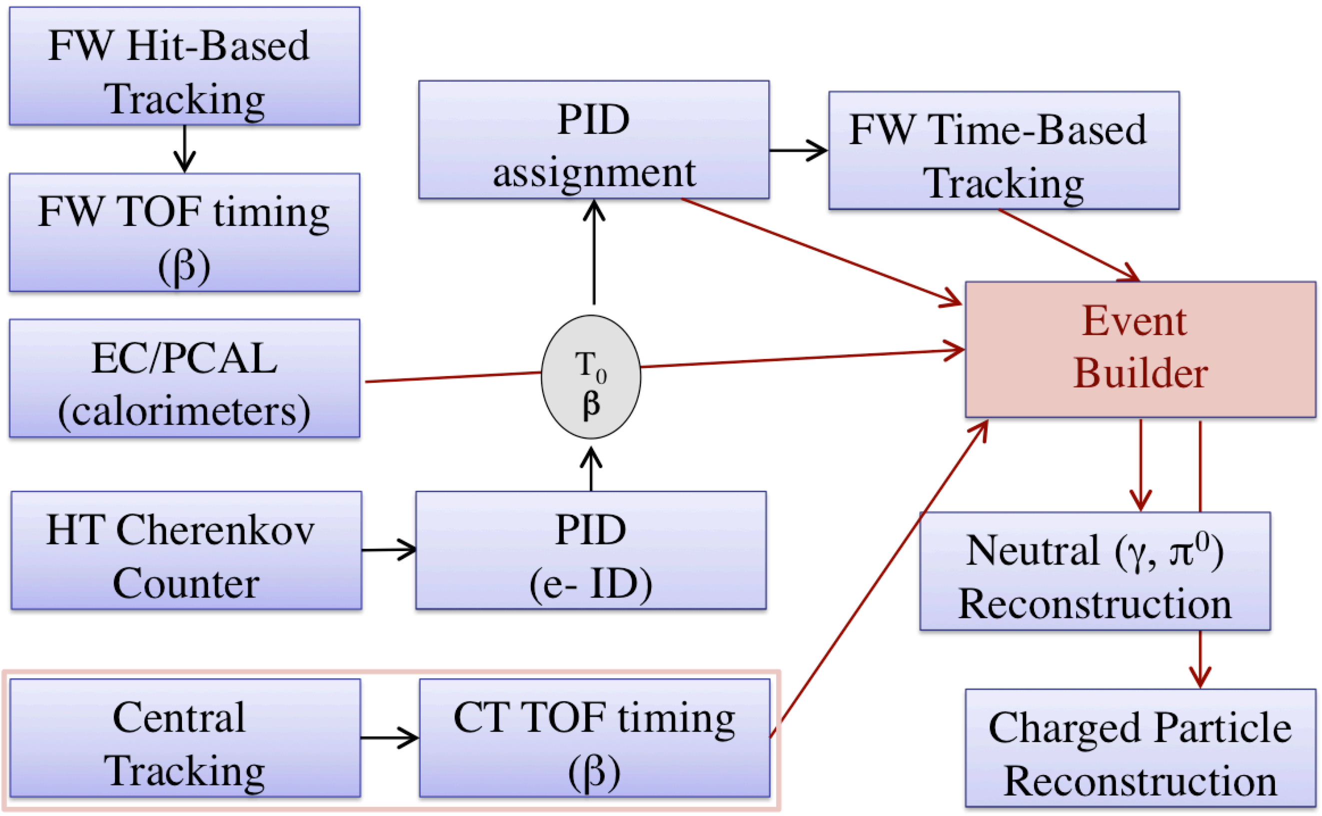
LTCC Reconstruction Software

- within CLAS12 framework in progress
- algorithmic procedure in place

Event Reconstruction Service Composition

- Each detector reconstruction component is a ClaRA service
- Overall Event reconstruction service (EB) combines info from service to reconstruct particle candidate

coatjava-3.0 service composition



Event Builder Development

J. Newton (ODU)

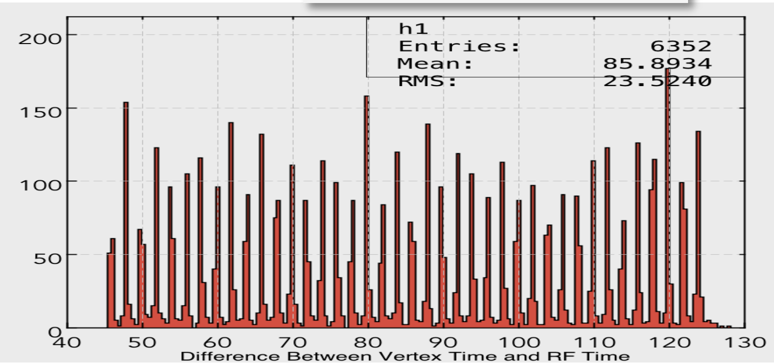


Fig. 1. This figure shows the difference between electron vertex time and the RF bunch time.

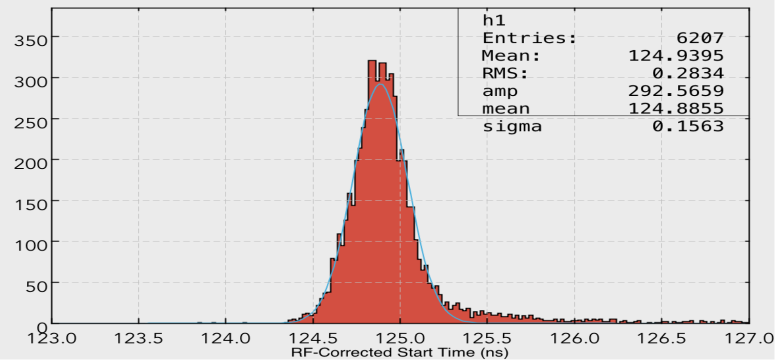
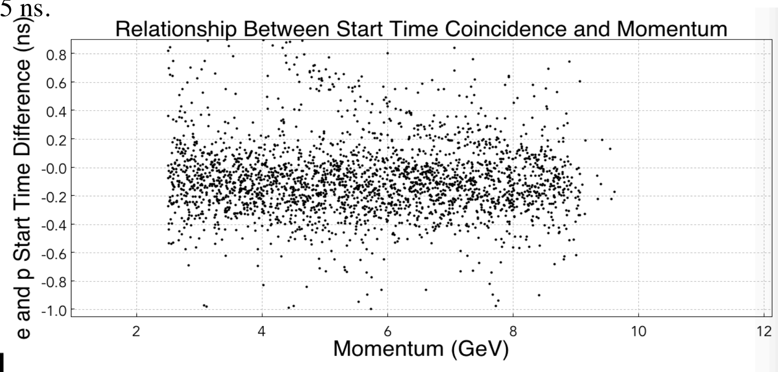


Fig. 2. Using the timing from FTOF1B, the electron vertex time is calculated and an RF-correction term is added to get a refined start time. For this simulation, the GEANT start time was at 125 ns.



Completed Features

- Matching of tracks in the drift chambers with the detectors in the forward detector
- Identification of a trigger particle
- Uses trigger particle vertex time along with RF time to establish event start time, which allows for calculation of speed of tracks and time-based tracking

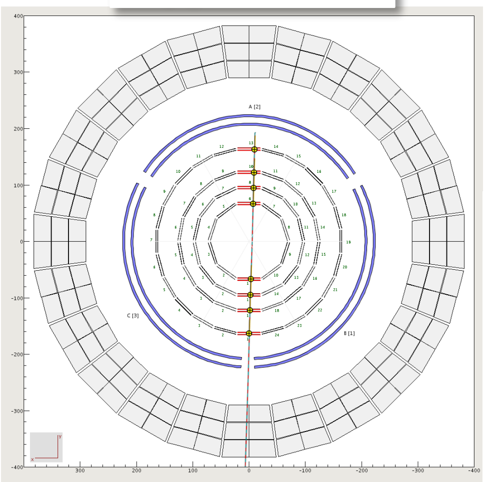
Works in Progress

- HTCC matching with cross positions at the edge of solenoid field
- PID Testing as soon as EC/PCAL is up and running

Alignment of the SVT using Millepede

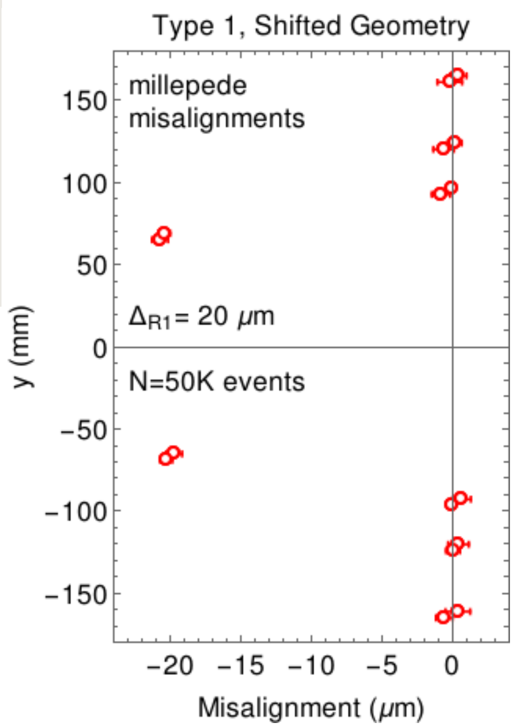
J. Gilfoyle

Type 1 track

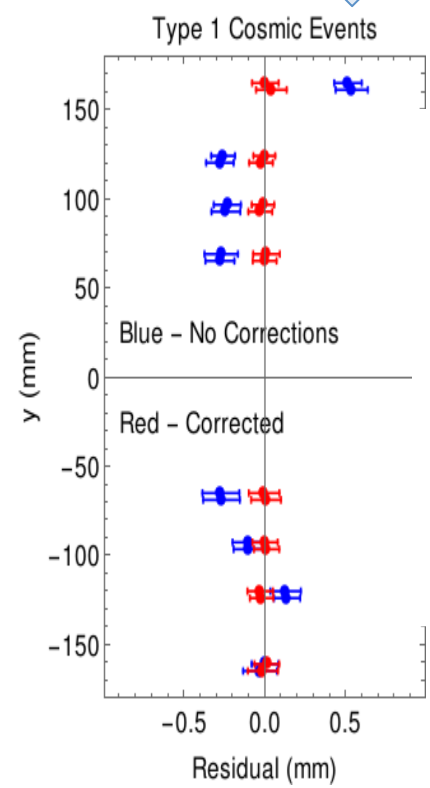


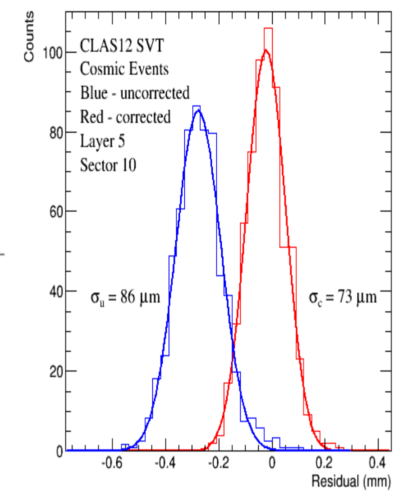
shift barrel R1 by 20 microns

Misalignment results obtained from Millepede tested on MC



Reconstruction of cosmic events with millepede misalignments incorporated





MC validations

validations using data

CLAS12 Calibration Constants

Unique storage of CLAS12 calibration constants is the mysql database ccdb (calibration constants database) developed by Hall D

https://clasweb.jlab.org/wiki/index.php/CLAS12_Constants_Database

Policies:

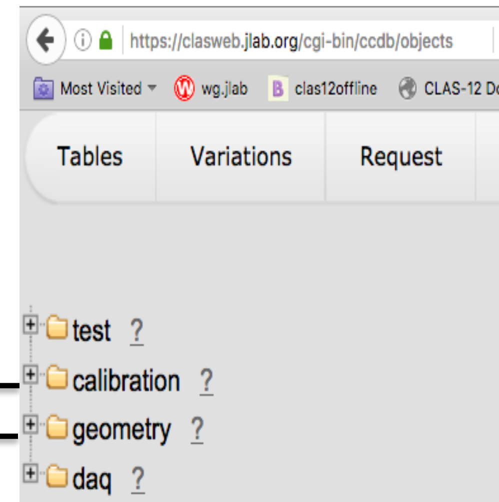
- use the same set of constants in GEANT simulation (gemc) reconstruction (coatjava), calibration software
- *lock* the constant set during the "production"

Basic Information about the CLAS12 Constants Database

mysql server host: clasdb.jlab.org
user: clas12reader
database: clas12
web viewer: https://clasweb.jlab.org/ccdb

Structure of clas12 db:

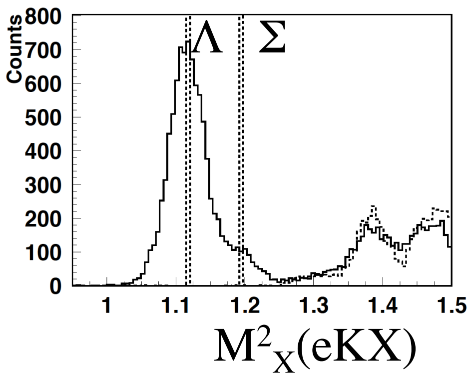
calibration constants
geometry parameters

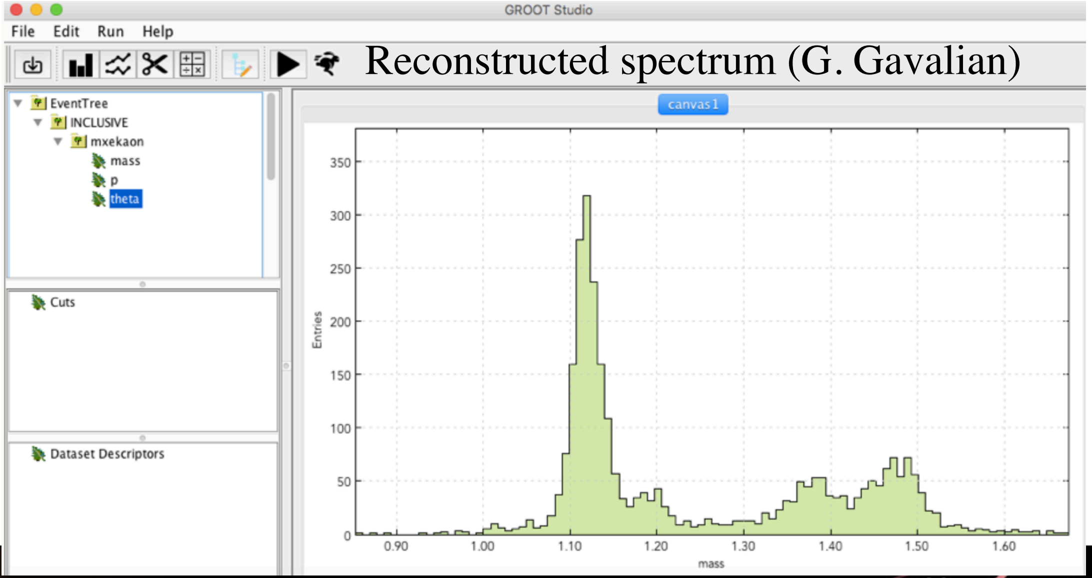


Release Version 3.0

- **coatjava-3.0 released**
- Comes with visualization tools and GUIs
  - Data Visualization package for plotting and fitting (using jMinuit)
    - Tuple tree implementation and output to compressed data format (HIPO).
  - Studio UI
    - Analysis studio for data analysis with interactive tools (fitting, custom functions, ASCII tuple import/export).
- Demo at 11/1/16 Workshop:
  - demo material by Nathan Harrison https://www.jlab.org/indico/event/180/session/13/contribution/99/material/slides/
  - Data analysis Studio demo by Will Phelps (FIU)

FastMC (H. Avakian)





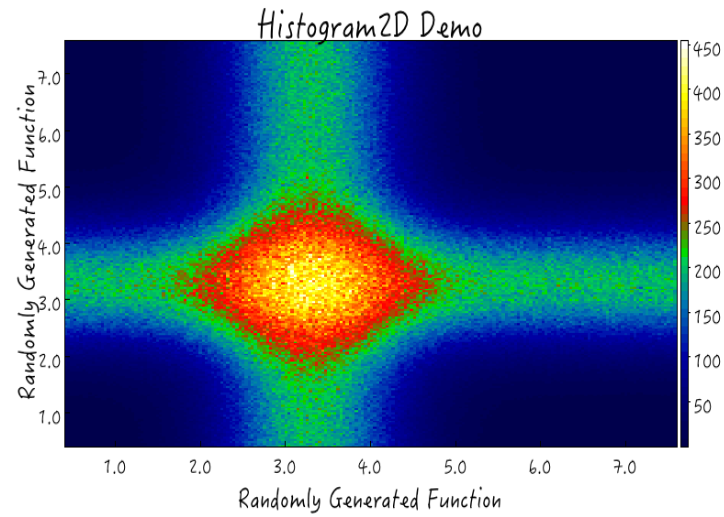
Reconstructed spectrum (G. Gavalian)

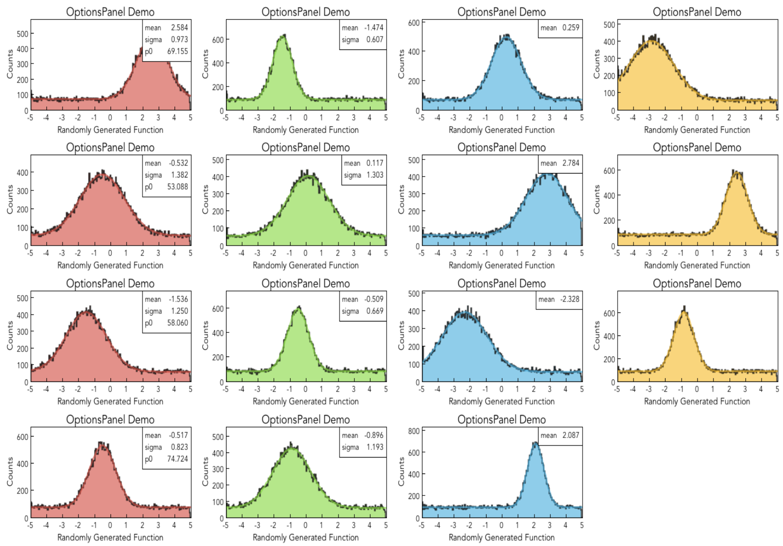
Common Tools (Plotting package)

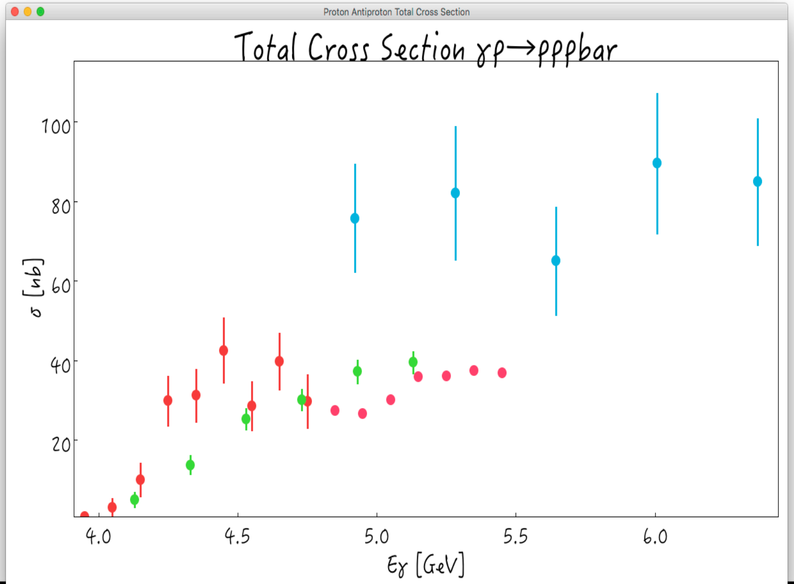
11/1/16 CLAS12 Software Demonstration

W. Phelps

- Histogram 1D/2D
- GraphErrors
- Fitting Package
- User defined functions
- GUI Tools





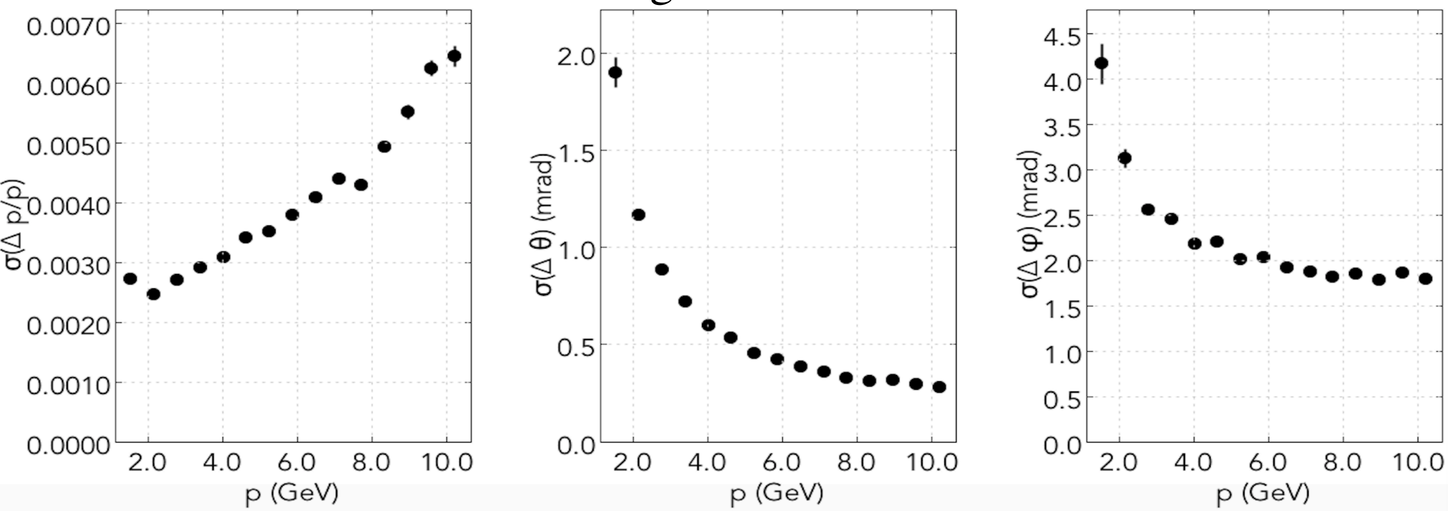


Jefferson Lab

Event Reconstruction Validations

N. Harrison

e- tracks generated at $\theta = 15^o$



Design Specs:

- $\sigma(\Delta p/p) < 1\%$
- $\sigma(\theta) < 1$ mrad
- $\sigma(\phi) < 3$ mrad

Tracking Efficiency:

$10^o<\theta<35^o$, p>1 GeV/c — DC

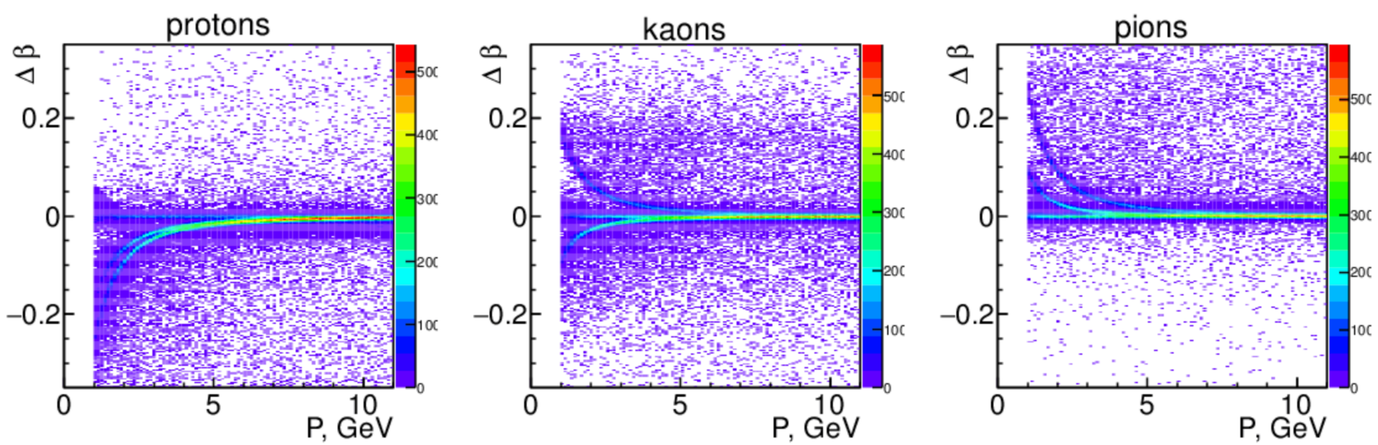
$\varepsilon > 95\ \%$

$45^o<\theta<110^o$, 0.5<p<2 GeV/c — SVT

$\varepsilon > 90\ \%$

PID [TOF validation by E. Golovach (Moskow State)]

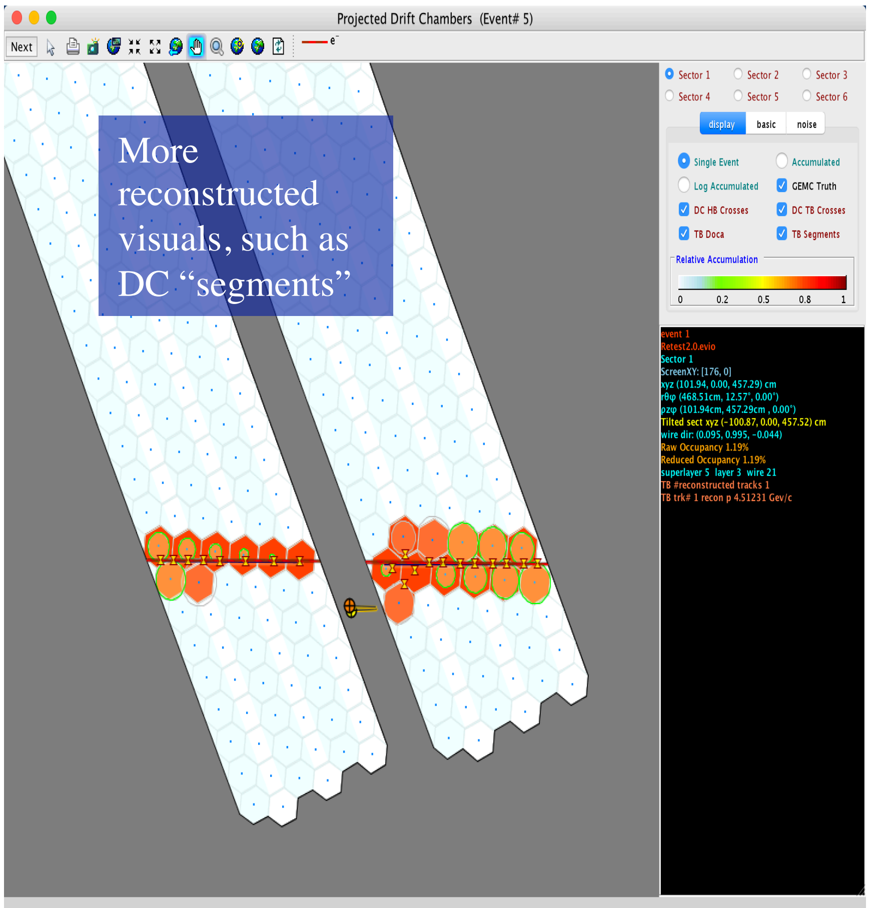
$\Delta\beta = \beta_{calculated} - \beta_{reconstructed}$
linear scale



Clas Event Display (*ced*)

D. Heddle

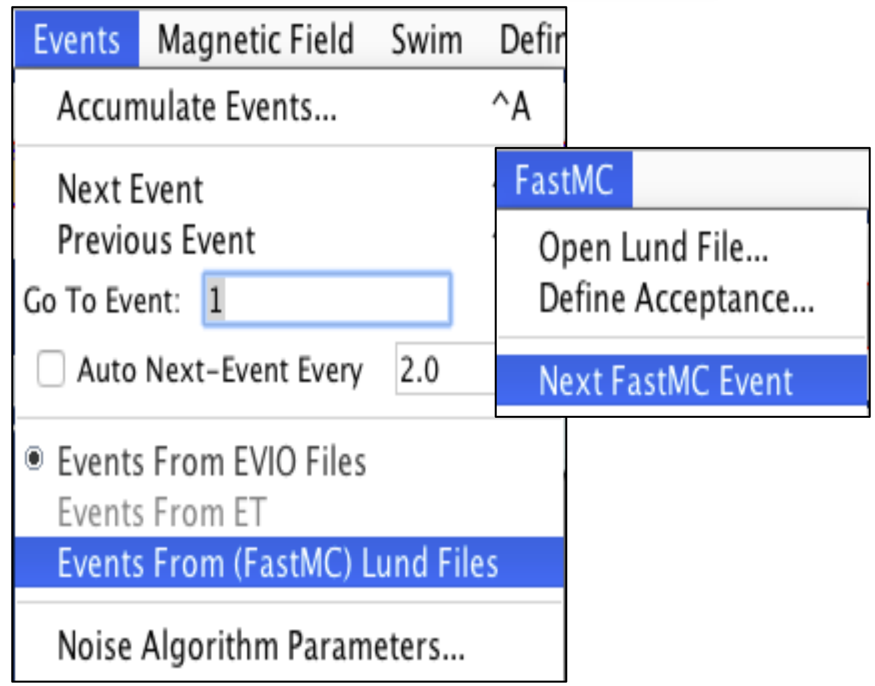
- **Newest addition HTCC**
- More views : "Projected Drift Chambers" view



migrated to coatjava-3.0

https://userweb.jlab.org/~heddle/ced/builds/

Full integration with FASTMC*, dictionary building, machine learning



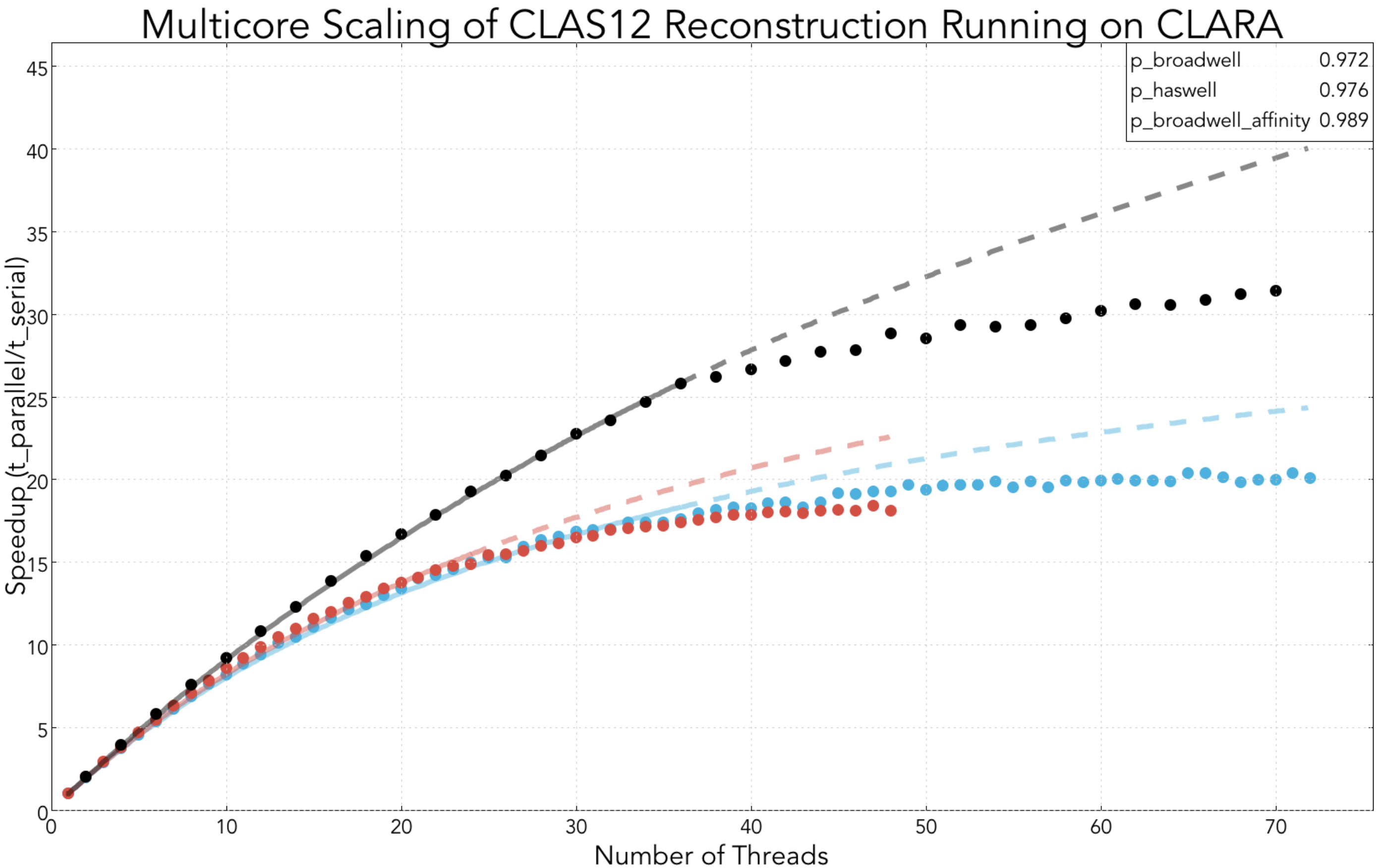
"Lund Files" as the source of events. *ced* will read the file and swim the particles.

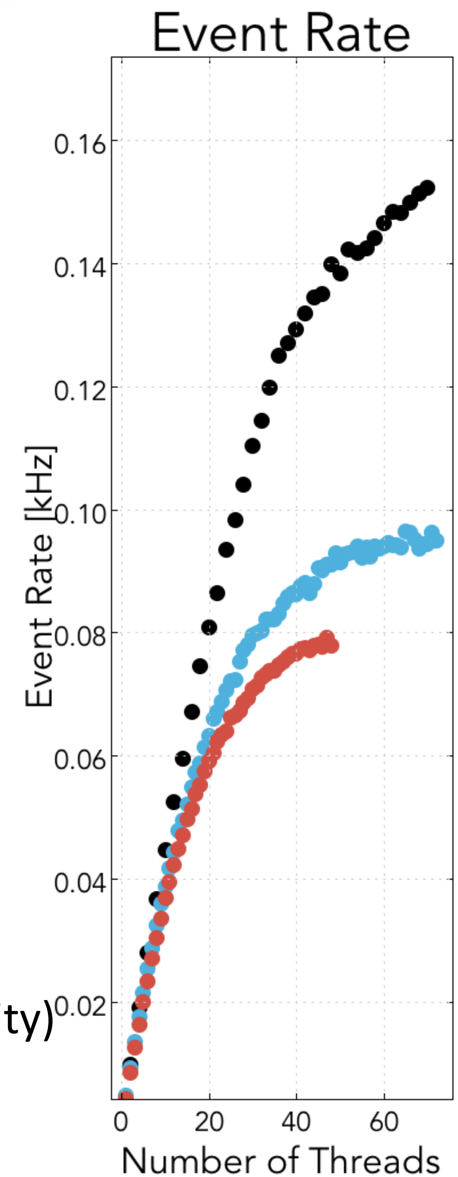
* FASTMC – produces detector hits by swimming simulated track in B-field

Code Management

- git repository
  - bug tracking
  - teams
- exploration of code analysis tool such as *coverity*
  - *using bug finding tool FindBugs*
  - validation suites in development
- Maven for version control & release

Event Processing Rates Analysis





- Tested on 3 different machines (broadwell, haswell, broadwell affinity)
- Up to ~99% scalability (Amdhal's law)
- Up to ~150Hz event rate / 2016 farm node

Data Formats and Preservation

- Transition to reading translation tables from ccdb completed.
- ADC pulse parameters read from database.
- Raw bank decoders implemented for all detectors. Transitional data structures are implemented (High Performance Output – hipo format) for data compression. Work is being done to optimize bank structures to save space.

Milestones

- 4th Quarter 2016
  - 3rd release including full event reconstruction available for detailed simulation studies, reconstruction of cosmic data, calibration and alignment code development
  - calibration suites in beta version
  - Calibration challenge Dec. 12—16, 2016
- 1st Quarter 2017
  - ◆ Code ready for KPP run
- 2rd Quarter 2017
  - Detector calibration code fully functional
  - missing databases in place
  - ◆ Code ready for engineering run
- 3rd Quarter 2017
  - ◆ Code ready for physics run
- 4th Quarter 2017
  - Code developed for physics analysis
    - Alignment and higher level analysis
    - validations using DVCS reactions and spin asymmetries measurements done on simulated data

Summary

- Framework
  - Common tools to facilitate development, uniform development environment with common geometry for reconstruction, simulation and visualization applications
  - High compressibility data format suitable for DST and data distribution
- Simulation
  - Geometry and digitization done for almost all detectors
- Event Reconstruction
  - Tracking complete for all baseline equipment
  - Event builder in place
  - Calibration suites advanced and ready for upcoming data challenge
- Visualization Tools
  - event displays, online monitoring suites in use
- Path to PRL
  - Reconstruction & Calibration software to be in place by start of data taking
  - ClaRA framework development for DST distribution

BACK-UPS

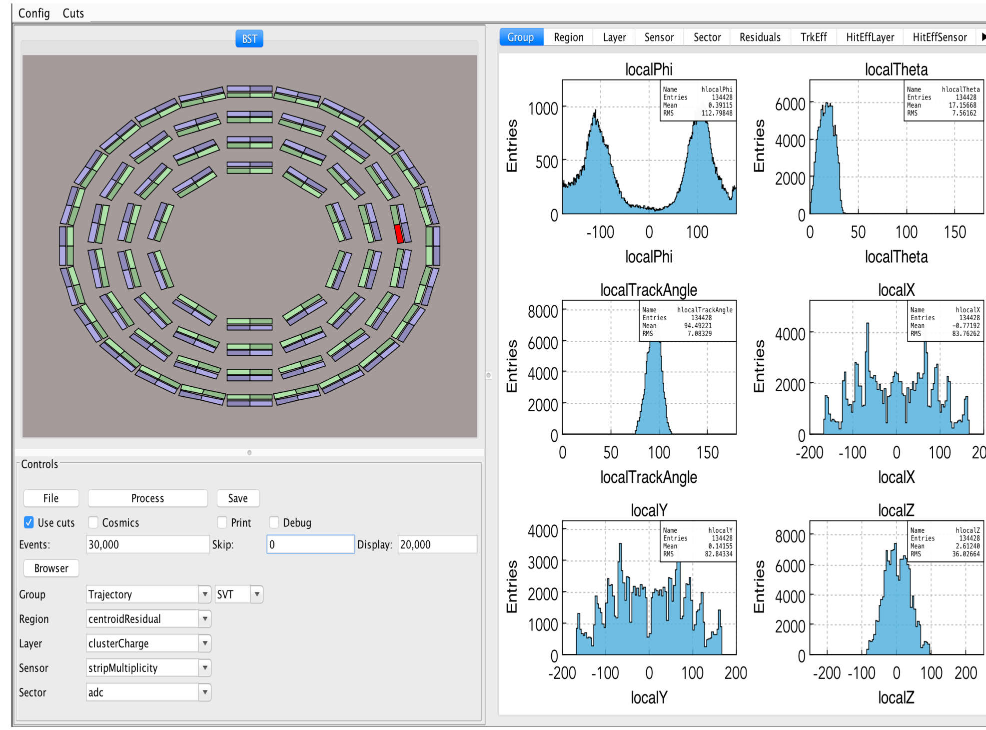
CVT Monitoring and Validation of Reconstruction and Simulation

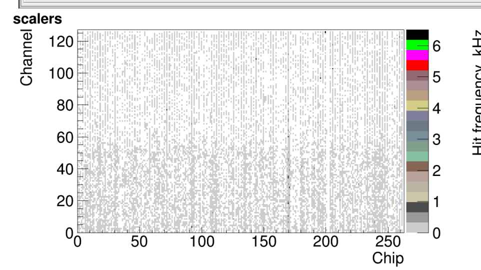
Monitoring
- Noise hit occupancy map implemented

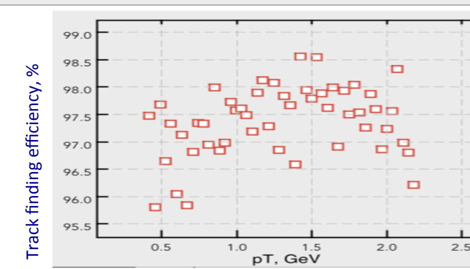
CVT Validation suite
- Histogram selection menus added
- MVT histograms added
- Cut selection menu implemented
- Event skimming added
- Efficiencies and resolutions implemented
- Hipo and root output format

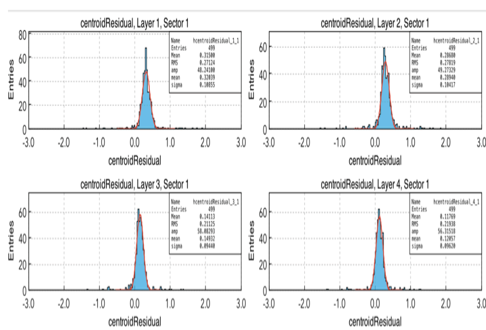
Validations performed
- Single track reconstruction
  - Geantinos, muons, pions
  - Straight (0T) and helical tracks
- Gemc 2.3
- Geometric acceptance
- Resolutions (momentum, angular)
- Efficiencies (track finding, hit finding)
- Occupancies

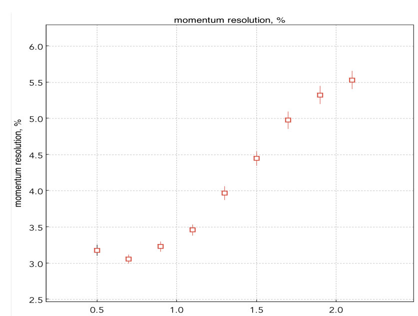
Work in progress
- Misaligned geometry
- Multiple tracks
- Electronic noise
- Local reconstruction
- Lorentz angle
- Documentation

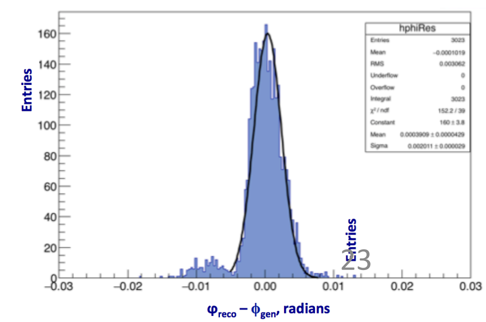












Background Rates

	DC Occupancy			HTCC Rates	
Field	**R1**	**R2**	**R2**	**nphe > 10**	**nphe > 30**
0	100	19.9	19.13	21,800	631
5	100	10.29	15.44	17,296	522
10	92.85	12.86	15.31	9211	136
20	38.06	6.94	9.23	309	1.1
30	15.94	2.41	4	129	1.8
40	7.08	1.21	2.07	87	1.5
50	2.41	0.84	1.53	18.7	1.3
60	1.63	0.66	1.36	10.9	1.4
70	1.89	0.79	1.07	9.7	1.1
80	1.51	0.63	0.97	7.3	1.4
90	1.14	0.64	0.95	9.6	1
100	1.45	0.51	0.84	7.7	0.5

Data Format
High Performance Output

HIPO Files (Gagik Gavalian)

As part of CLAS12 software framework HIPO library was developed for storing reconstruction output and possibly the final DSTs. HIPO provides random access to compressed data sets and has no limitation on file size. It is useful for chaining many evio files together to save on storage and have ability to process them at once.

Creating a Hipo File

There is a Hipo convertor provided with coatjava distribution. To combine multiple EVIO files into one HIPO file use the command.

```
>bin/hipo-writer output.hipo input1.evio input2.evio input3.evio
```

This will create a large file, reduced in size due to internal compression. The events stored inside are EVIO events, they are grouped together and indexed for easy and fast random access.

Reading Hipo Files

Reading Hipo files is not different from reading EVIO files. The DataSource objects are interfaced inside of our framework so they all behave in exactly same way.

```
import org.jlab.evio.clas12.*
import org.jlab.data.utils.DictionaryLoader;
import org.jlab.clas.tools.utils.*;
import org.jlab.hipo.*;

filename = args[0];

HipoDataSource reader = new HipoDataSource();
reader.open(filename);
int nevents = reader.getSize();

int counter = 0;
for(int i = 0; i < nevents; i++){
  EvioDataEvent  event = (EvioDataEvent) reader.gotoEvent(i);
  event.show();
  counter++;
}
System.out.println("  procecessed " + counter + "  events");
```

$\gtrsim$ 50% compression & faster file reading/processing time

Recommendations

- Explore the use of Analysis Trains in collaboration with GlueX, so the technology is in place once the data become available.
  - Halls B & D are adopting plugin model. Hall D has a working model for analysis trains for monitoring, reconstruction and analysis.
  - Hall B plan to adopt a similar approach to data processing using clara modularity and multithreading.
  - A combined effort is planned for data cataloguing and distribution.