CLAS12 Offline Software Tools

G.Gavalian (Jlab)

Overview

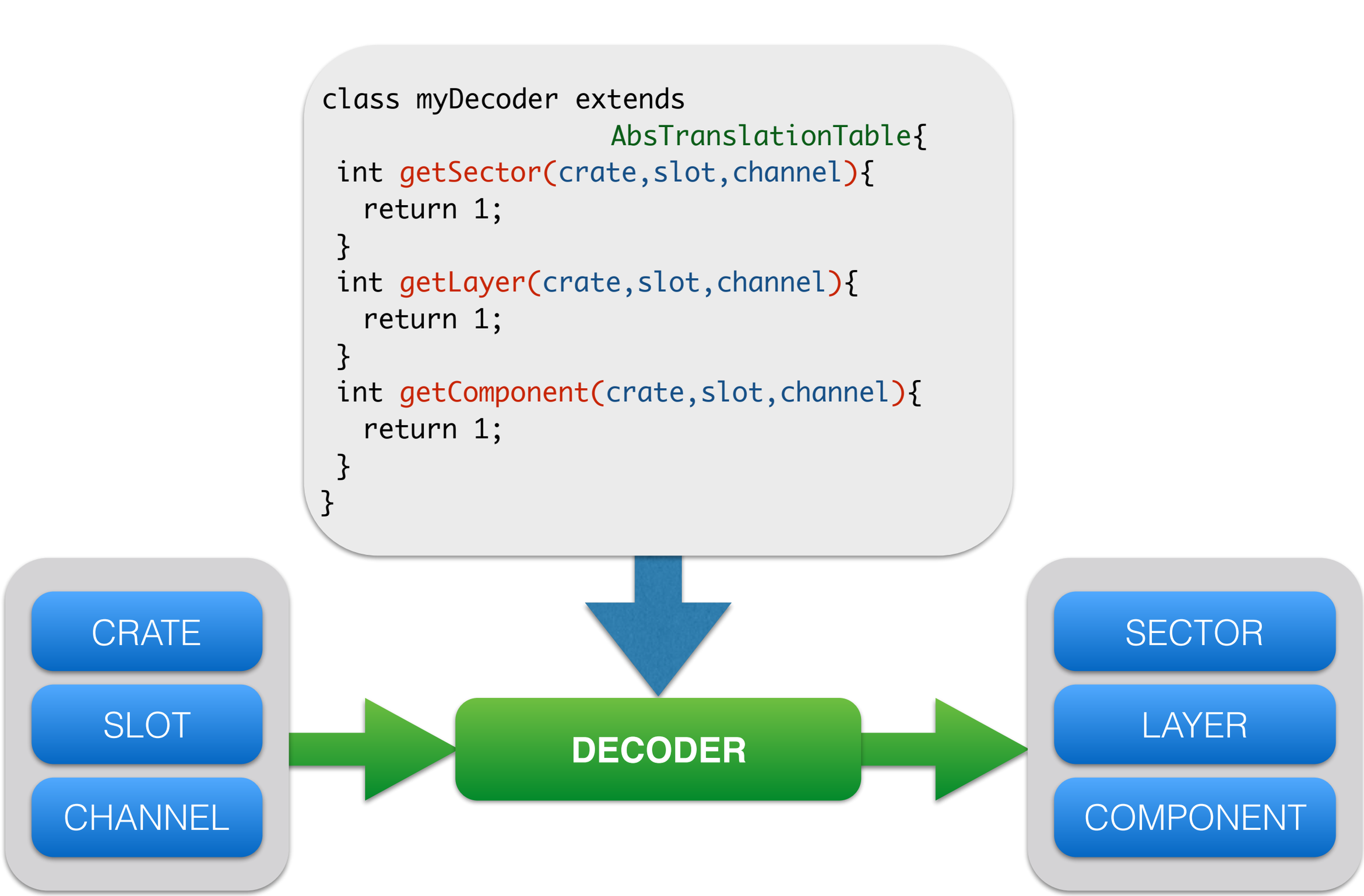
- **Data formats I/O**
  - gemc data reader
  - raw data reader
  - detector hit decoder
  - ET ring data reader
- **Geometry Package**
  - implementation of all baseline detectors
  - 3D viewer in CED
- **Plotting Package**
  - modern look and feel
  - fitting with Minuet
  - Latex label and text support
- **Calibration software**
  - unified calibration and monitoring suite
  - geometry tied to calibration plugins

EVIO Data Format

- ✓ **Dictionary Based I/O:**
  - reading banks produced by GEMC
  - writing reconstruction output banks
  - Dictionary in the file
- ✓ **Raw Data I/O:**
  - reading coda data for different modes (MODE=1,3,7)
  - automated translation tables for detectors (plugins)
  - standardized hit bank generation
- ✓ **EVIO utilities:**
  - splitting and merging files
  - GUI for viewing GEMC generated banks (in CED)
  - GUI for viewing and fitting RAW ADC spectra
  - Reading data from ET ring (online)
- ✓ **BOS utilities:**
  - Reader/Convertors for BOS (CLAS6) data to EVIO format
  - Unified Interface for analyzing data from CLAS6
  - Framework for PID, cuts and corrections for CLAS12 and CLAS6
  - Data format from CLAS6 can be passed to CLAS12 calibration and monitoring.

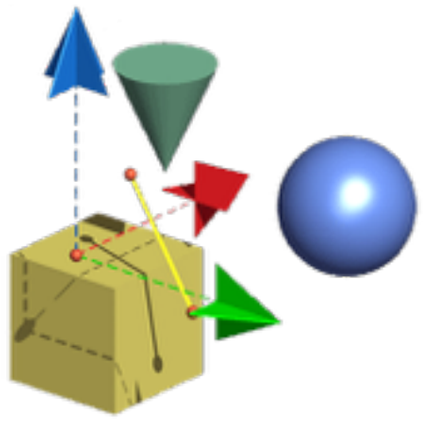
Raw Data Reading

```
class myDecoder extends
                    AbsTranslationTable{
 int getSector(crate,slot,channel){
   return 1;
 }
 int getLayer(crate,slot,channel){
   return 1;
 }
 int getComponent(crate,slot,channel){
   return 1;
 }
}
```



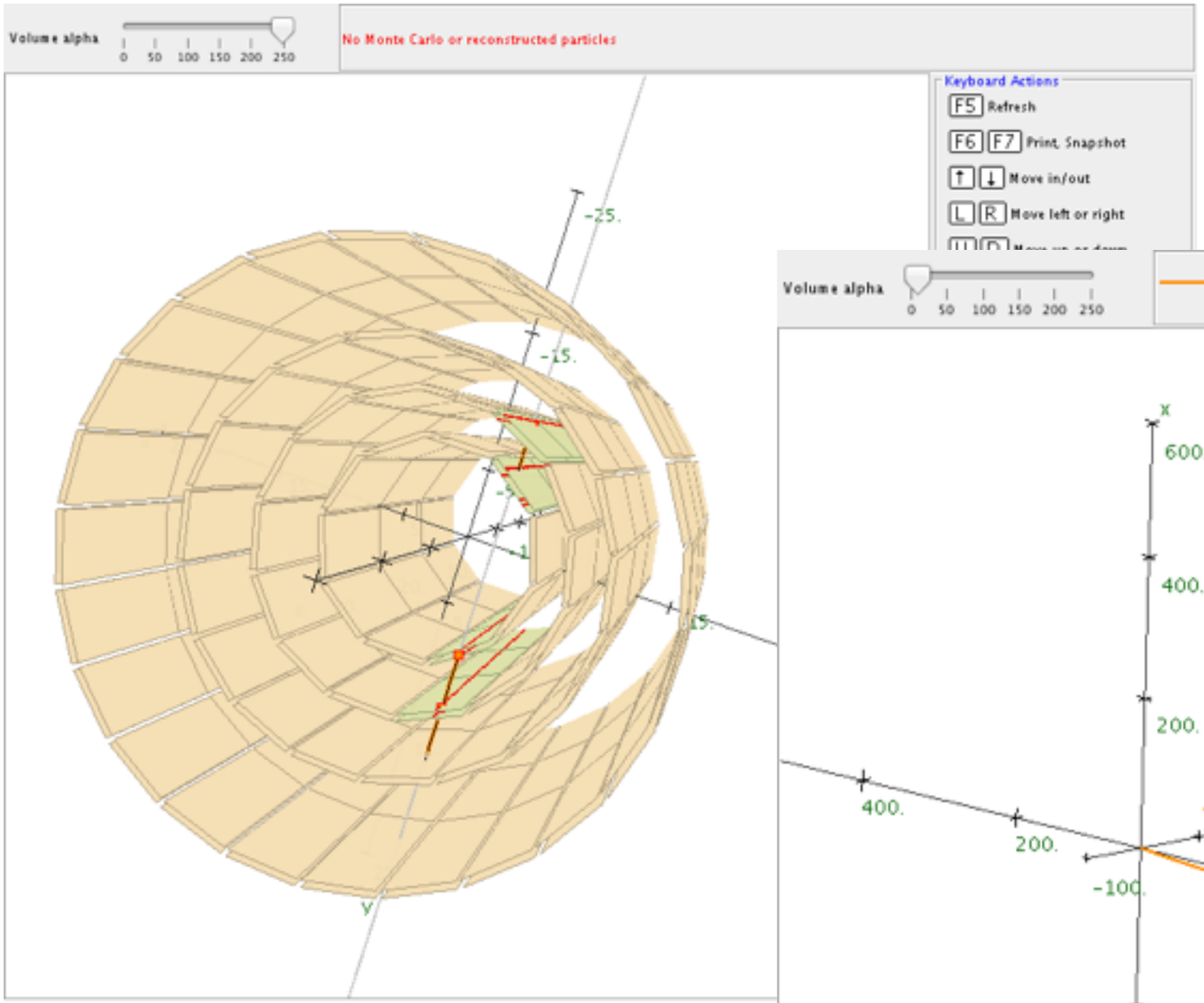
Geometry

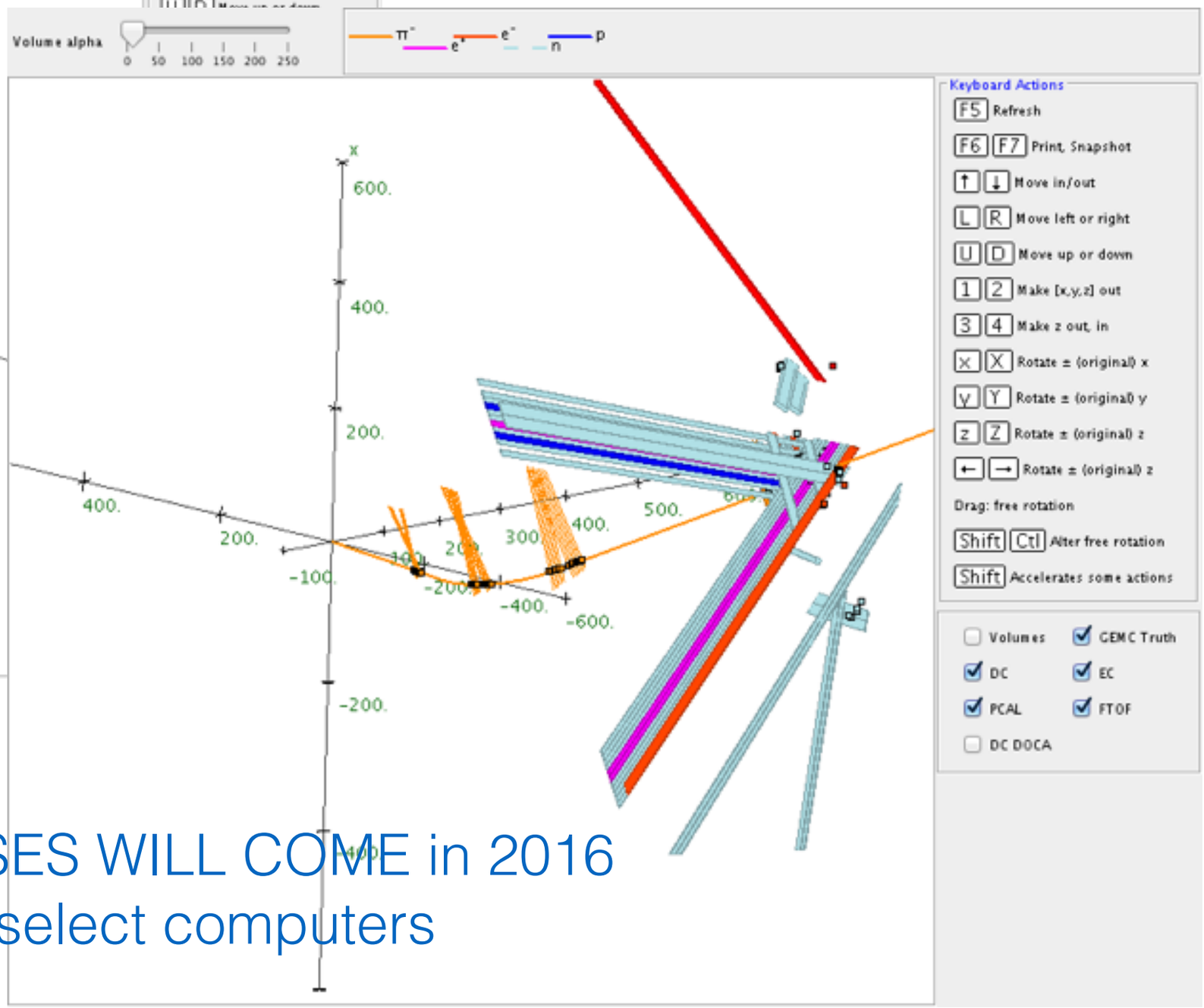
- ✓ **Standart Detector Geometry Package Implements:**
  - Forward Time of Flight
  - Electromagnetic Calorimeter
  - Forward Tagger
  - Drift Chambers
  - Silicon Vertex Tracker
  - Central Neutron Detector



- ✓ **Geometry Tools and Utilities:**
  - Drawing package for 2D detector representation
  - 3D shapes for CED-3D viewer
  - Detector component tracker for Fast Monte-Carlo

Geometry 3D in CED

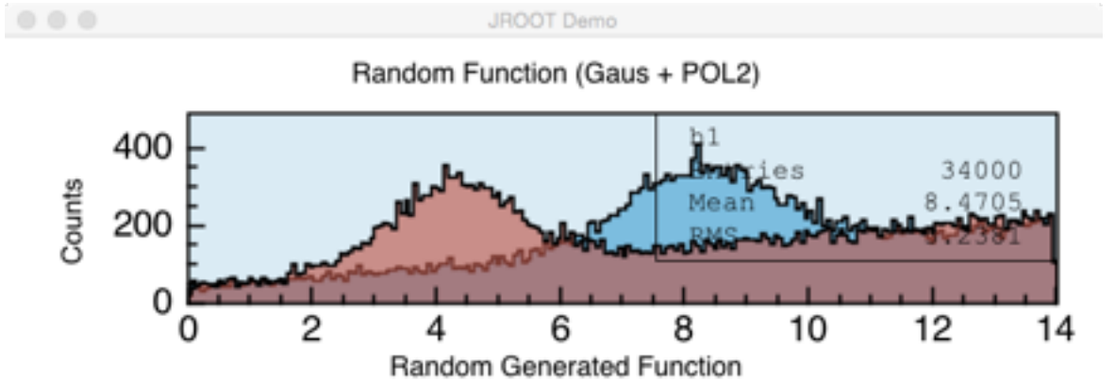


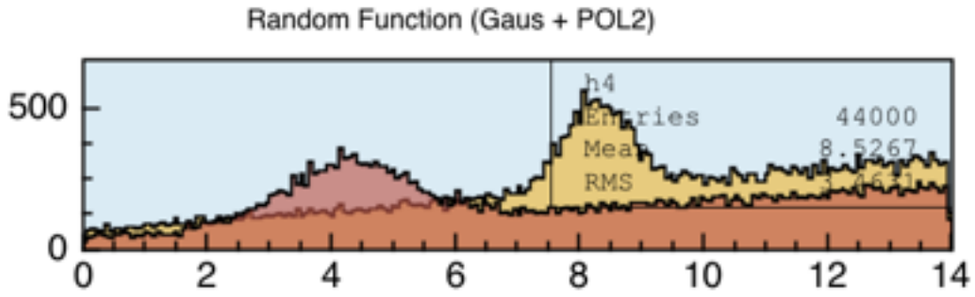


3D GLASSES WILL COME in 2016
on select computers

Data Visualization

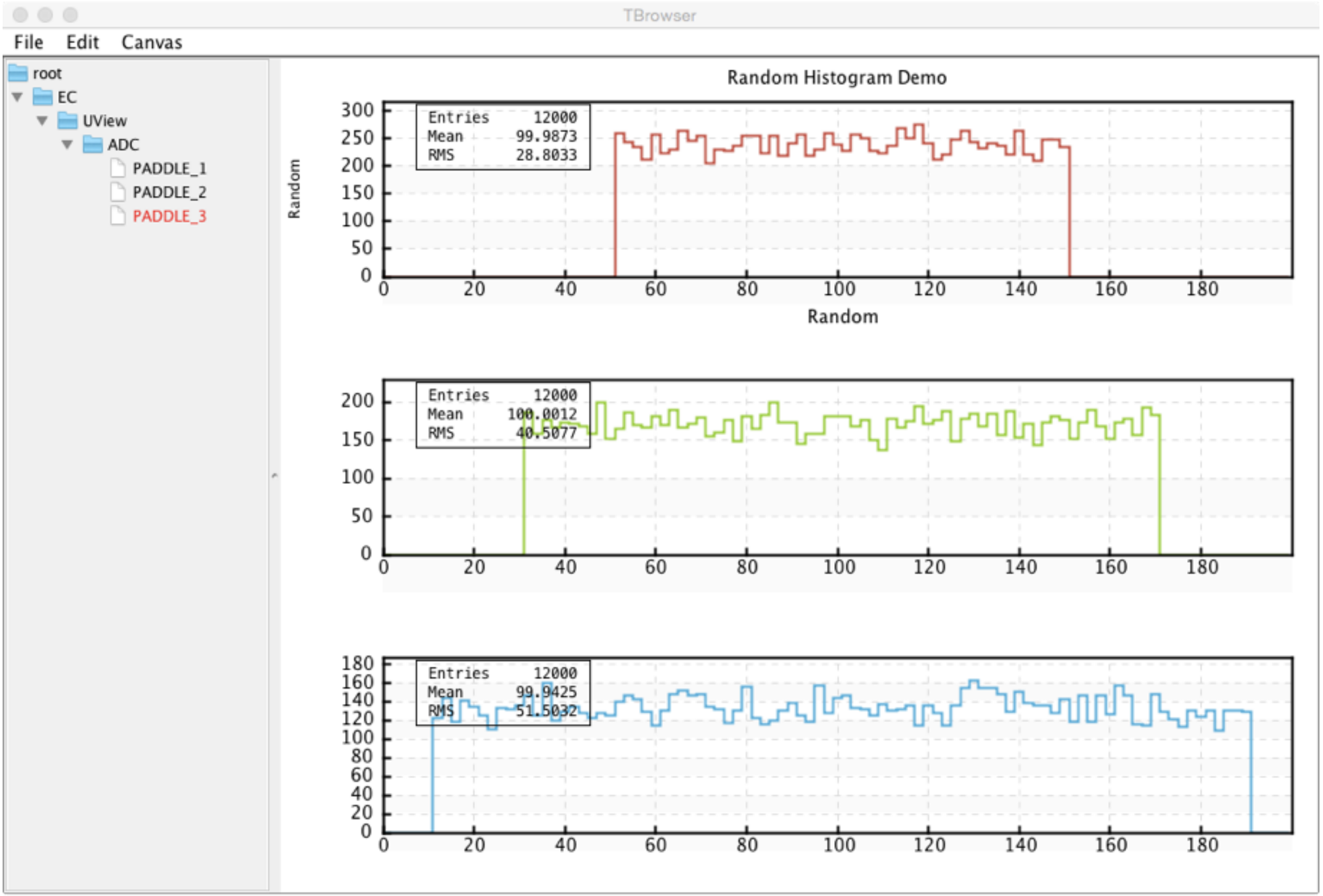
- **Plotting Library:**
  - histogram and graph classes implemented
  - plotting implemented matching ROOT API
  - Fitting added using Minuet library
  - Latex parsing added for titles and texts.
  - I/O of histograms implemented, with browser.
  - TNtuple class implemented for cut based plotting.
  - NTuple I/O implementation in progress



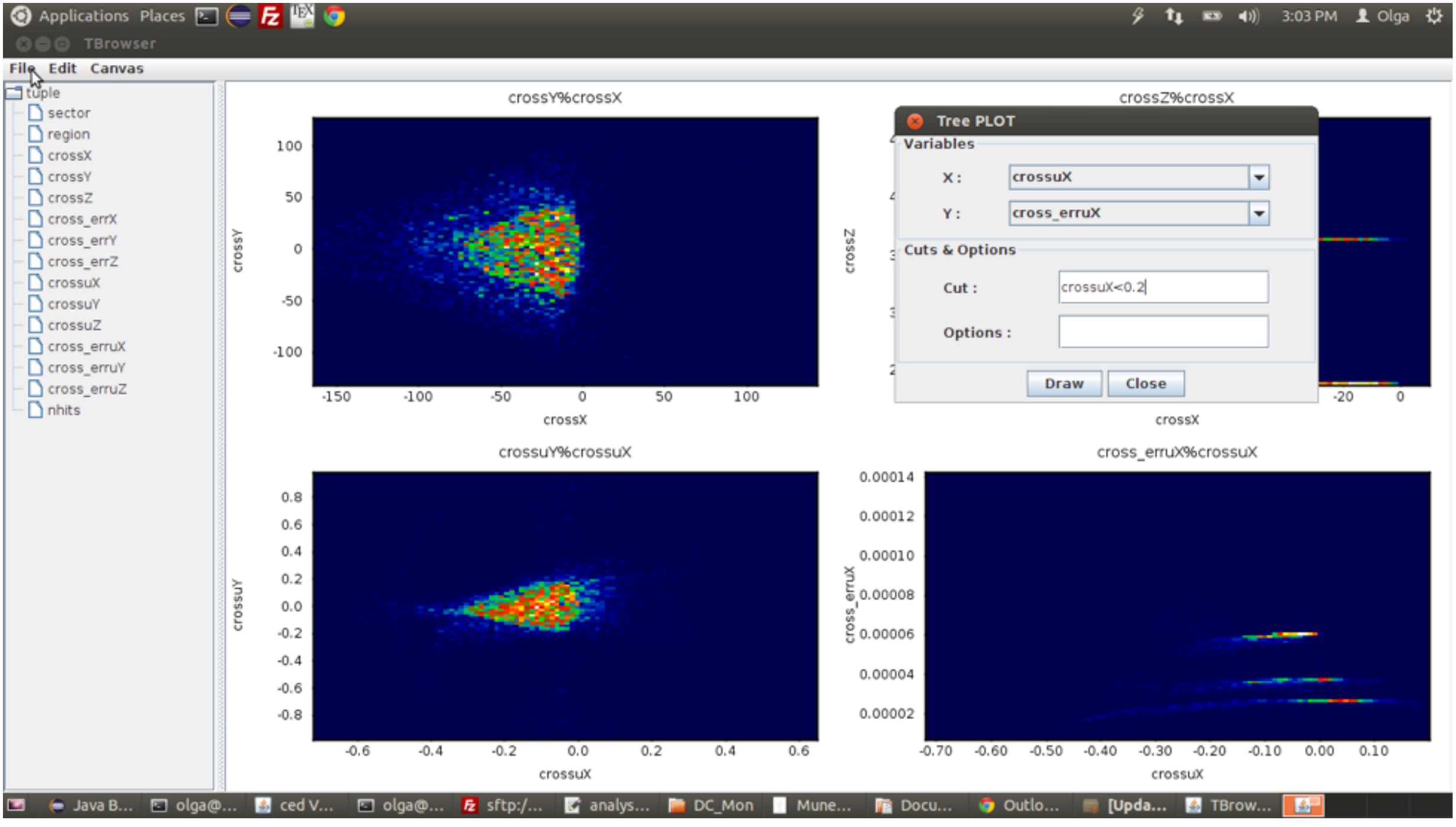



```
14  TGCanvas c1 = new TGCanvas("c1","JROOT Demo",900,800,1,1);
15  //c1.setFontSize(14);
16
17  H1D h1 = new H1D("h1","ep #rarrow ep#gamma",200,0.0,14.0);
18  H1D h2 = new H1D("h2","ep #rarrow ep#gamma",200,0.0,14.0);
19
20  h1.setXTitle("M^2 [GeV^2]");
21  h1.setYTitle("Counts");
```

Histogram Object Browser



Tree Browser Object (DC monitoring)



Calibration and Monitoring

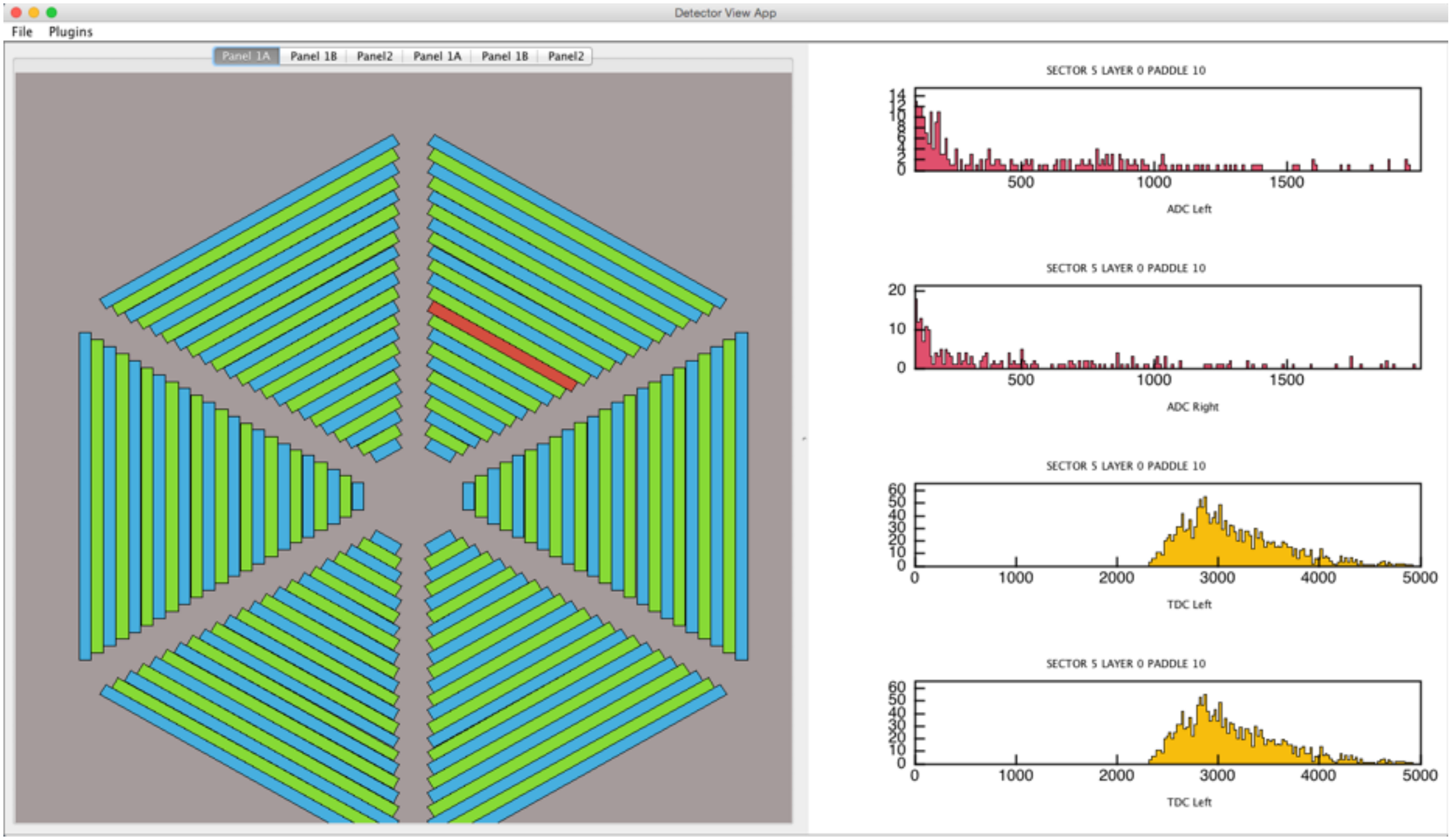
- ✓ **Calibration & Monitoring Software:**
  - Plugin based software framework
  - standard interface for passing data through modules
  - standard representation of the detector components
  - interface to draw relevant histograms for each component
  - automated plugin discovery from the package

void processEvent(EvioEvent e)

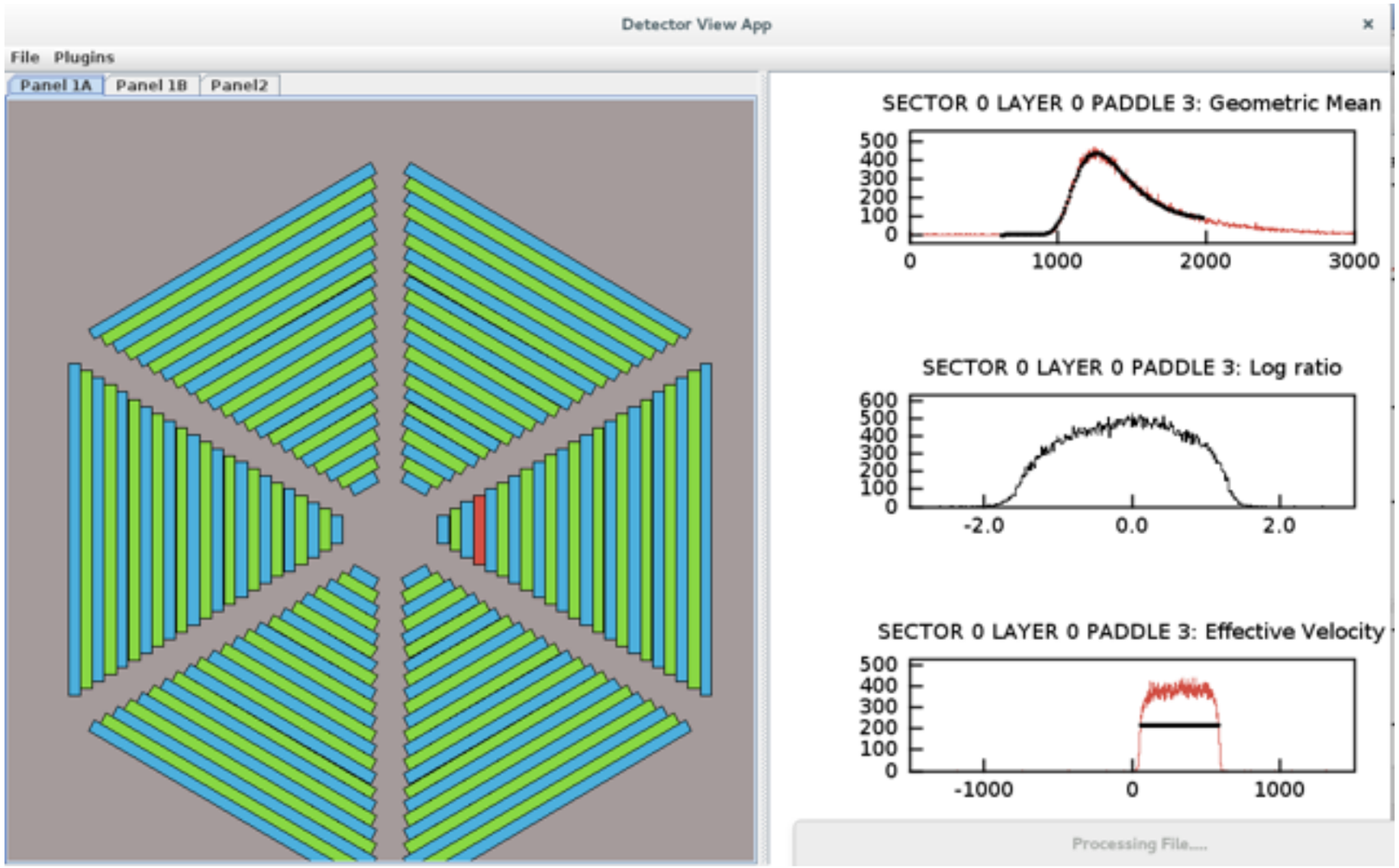
void drawComponent(sector, layer, component, canvas)

void getColor(sector, layer, component)

Calibration and Monitoring



FTOF Calibration



Calibration area	COATJAVA development status
Geometric mean / Log ratio	*In progress*
Effective velocity	*In progress*
Attenuation length	*In progress*
Time-walk	*Planned*
Counter status	*Planned*
TDC	*Planned*
RF offset	*Planned*
P2P constants	*Planned*

Work in progress

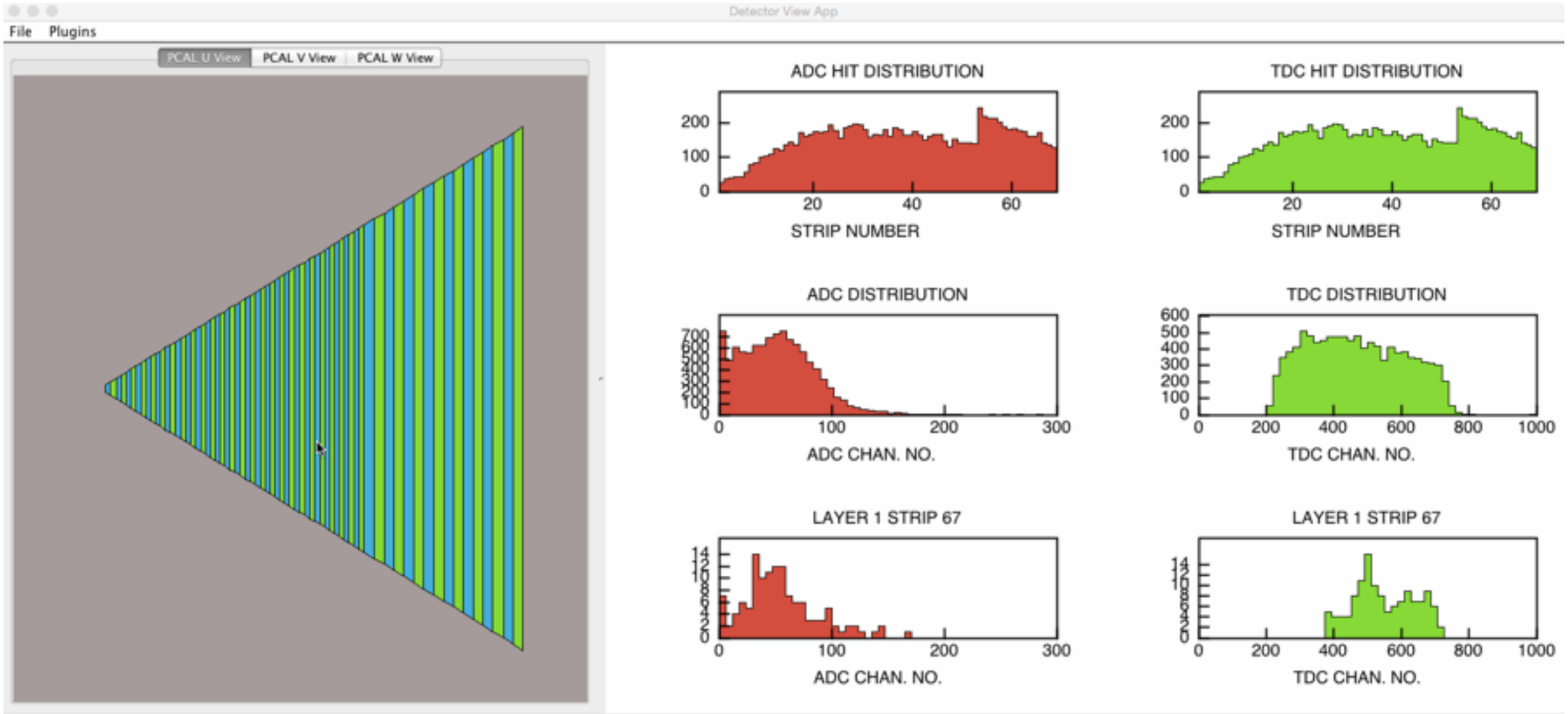
- Conversion of calibration algorithms to COATJAVA framework
- Integration of calibration plots and fits with standard monitoring GUI

Work planned

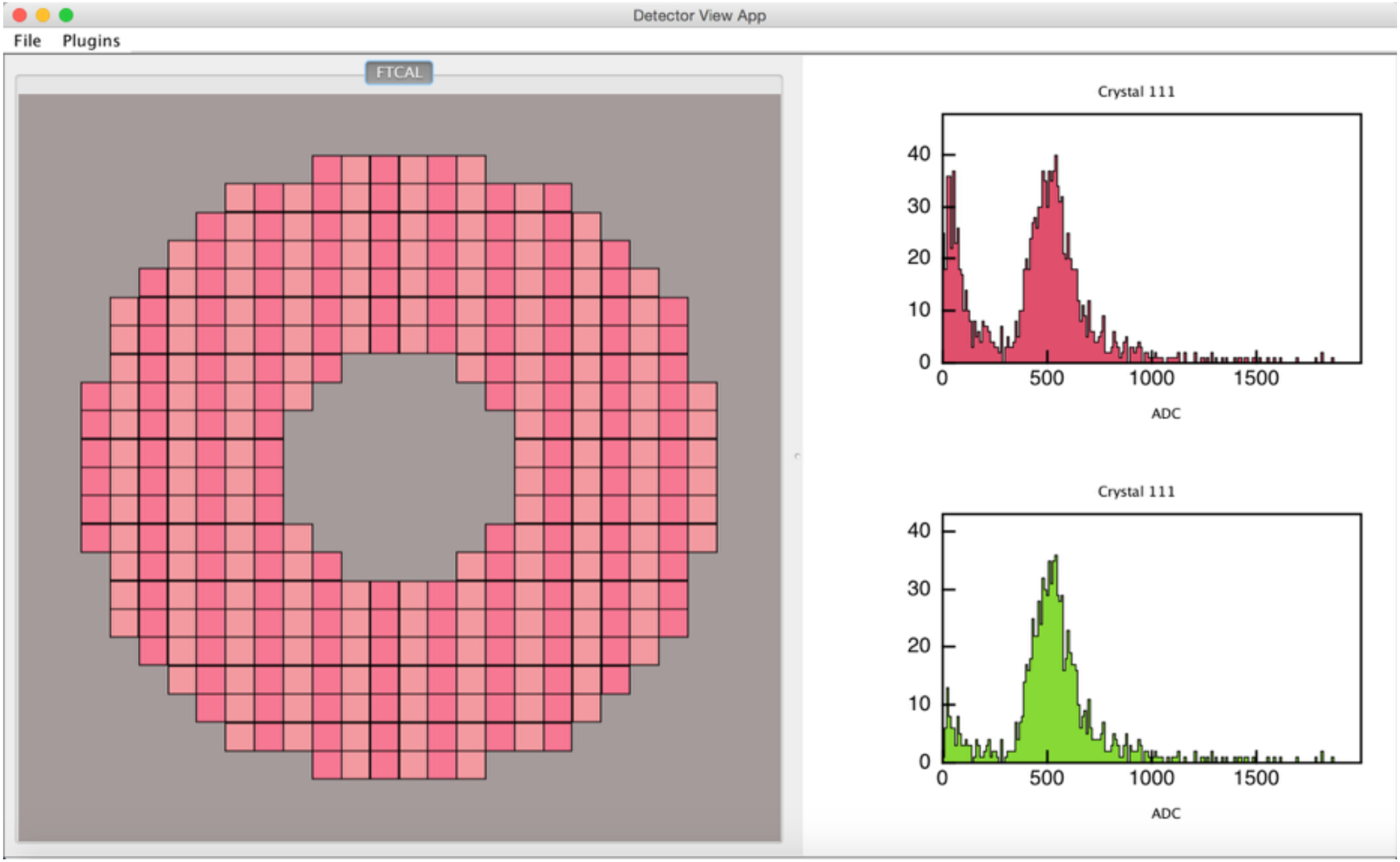
- Conversion of remaining calibration algorithms
- Fuller functionality within GUI and interfacing to calibration database

PCAL Calibration

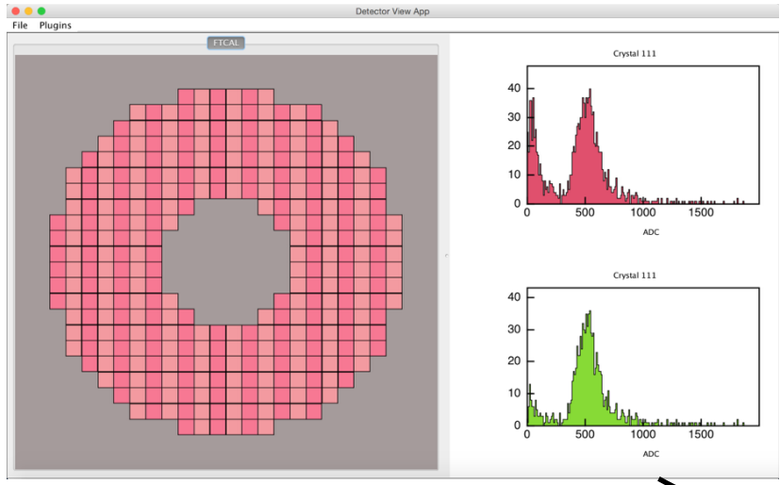
- ✓ **PCAL/EC Calibration Software**
  - switched to using common tools
  - interfaces with UI developed for all detector components

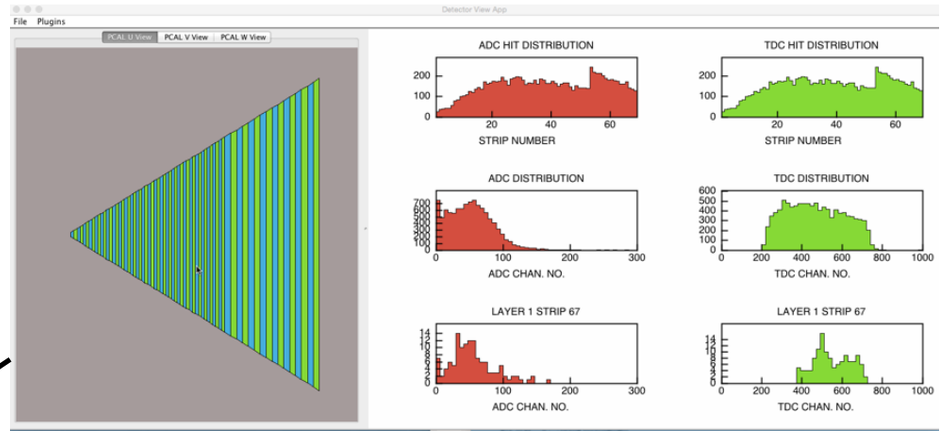


FTCAL Calibration

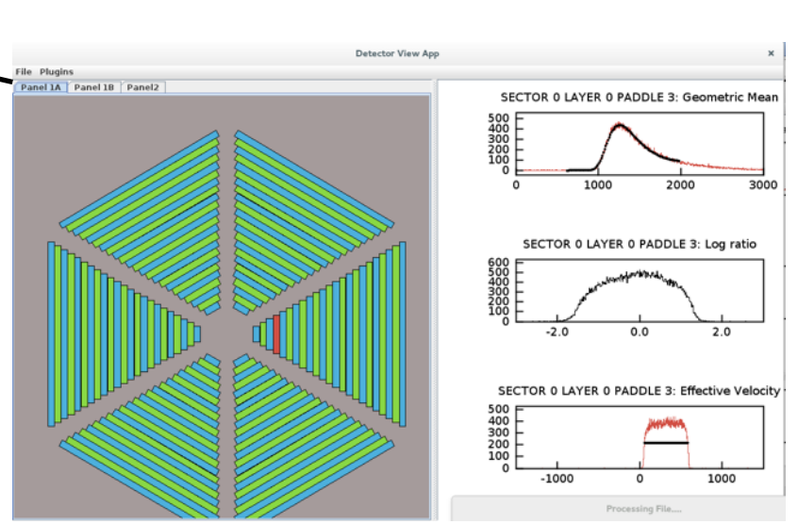


Commissioning





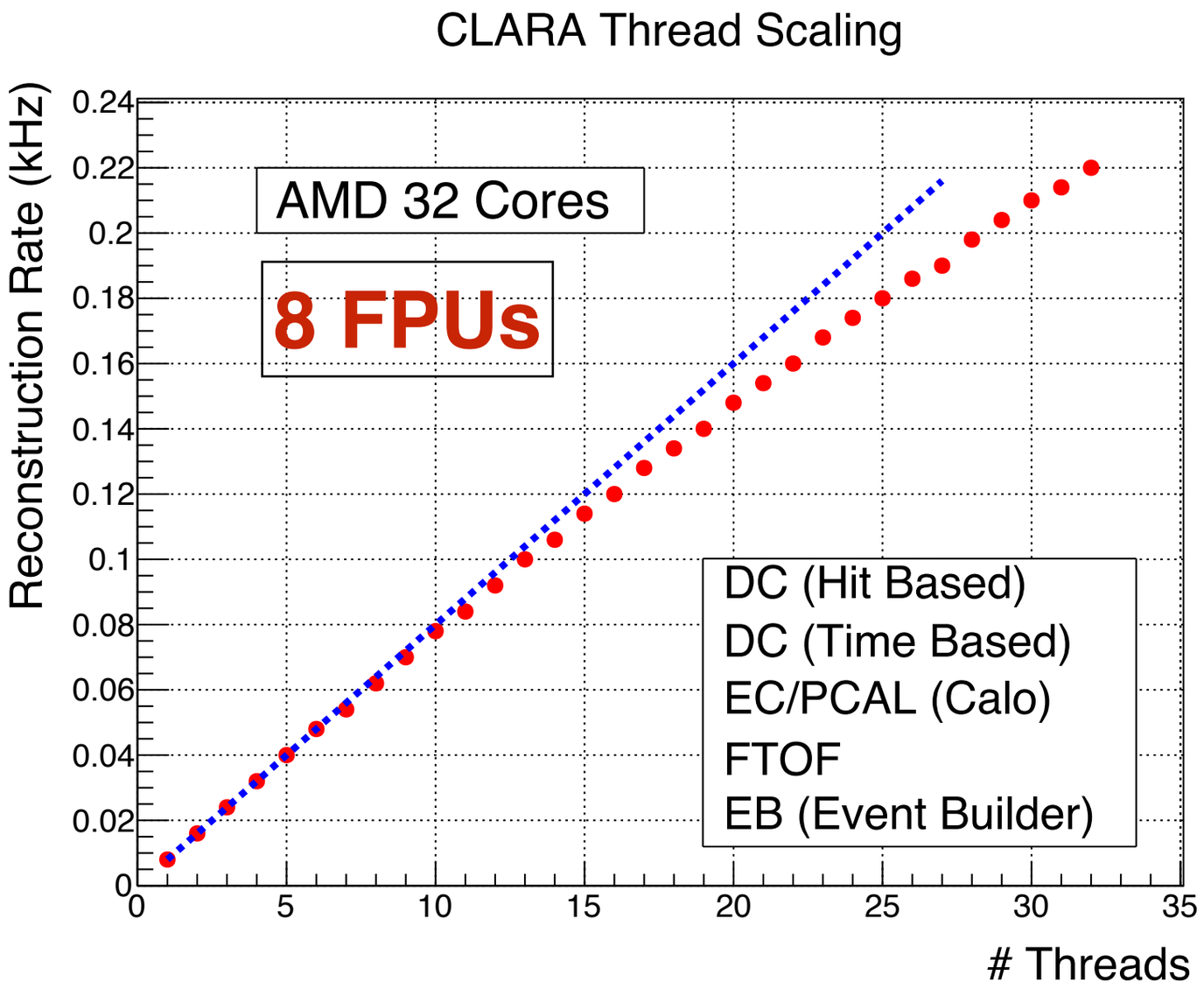
ET
(DAQ)



- ✓ **Calibration & Monitoring Software:**
  - ET ring data reader in I/O framework
  - Interface to run Calibration and monitoring software from ET ring.

CLARA Data Challenge

- Reconstruction code tested in cloud environment (CLARA)
- Full chain running on one 32-thread (16-cores) machine
- Scaling with cores is linear
- Scaling with threads behaves as expected
- Reconstruction runs 6 ms/event on 24 core Haswell (50 machines to keep up with DAQ)



Documentation

- ✓ **CLAS12 Offline software documentation:**
  - located : http://clasweb.jlab.org/clas12offline/docs/software/html/

- ✓ **Documented Software components:**
  - EVIO I/O for GEMC data and Raw DAQ data (good)
  - Geometry package usage (fair)
  - Plotting package (in progress)
  - implementation of detector reconstruction (good)
  - calibration and monitoring software implementation (good)

What we need

- ✓ **Geometry:**
  - Detector groups to take ownership of the code and maintain it
  - Detector groups to come up with misalignment representation.

- ✓ **Calibration and Monitoring:**
  - Each detector needs to have a monitoring module developed.

- ✓ **Decoding:**
  - Translation tables need to be implemented for detectors
  - Common structure for writing EVIO files from composite format

- ✓ **Calibration DB:**
  - Define calibration constant tabes in the CCDB database
  - Database contact person for collaborators

Summary

- Data reading/writing routines are all in place for detector commissioning and reconstruction software.
- Convenient framework is developed for DAQ data translation.
- Reconstruction plugin abstract classes are finalized and are ready for users.
- Geometry package is mature and implements most of the baseline detectors. It is used throughout entire CLAS12 software package including CED.
- Monitoring abstract interfaces are implemented and being developed based on group recommendations. Many groups started using it.
- Plotting package is sufficient for displaying plots and analysis results, features are being added constantly.