Photo: Steven Saffi / University of Adelaide

Development of a standalone Data Quality Monitoring viewer

(Data Quality Monitoring AND Long Term Perf.)

Fred Sarazin (fsarazin@mines.edu)
Physics Department, Colorado School of Mines

LTP task call, July 26, 2018

Motivation

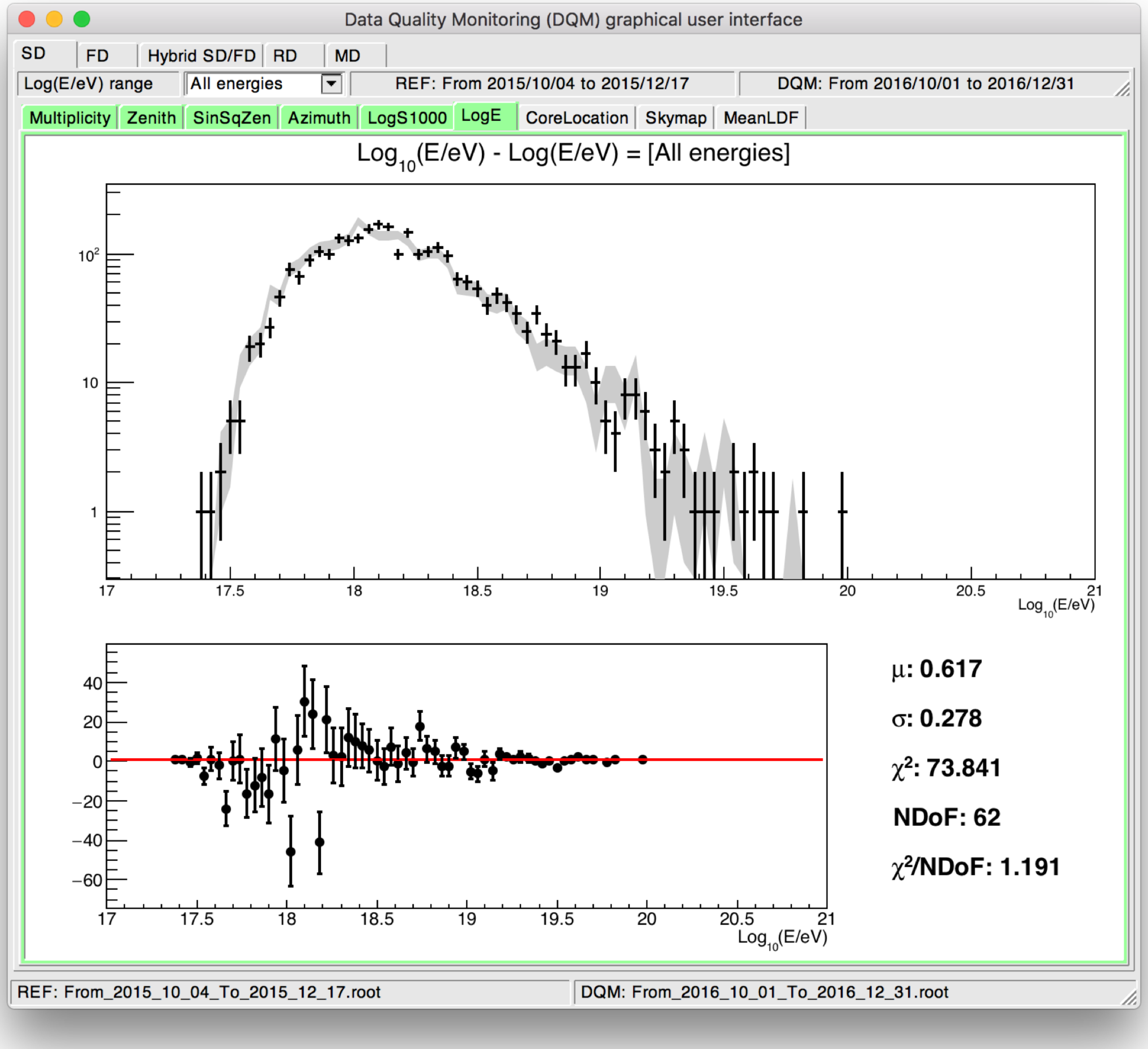
Data Quality Monitoring (DQM):

- Check the quality of the post-processed (merged) data – ADSTs
- Complementary to raw data monitoring
- Also a useful tool to start doing **long-term performance studies**

DQM viewer:

- First presented at KIT analysis meeting, June 2016
- Version 1 essentially ready for release – tested on Linux and Mac
- Downloaded / installed / tested by: Corinne / Bruce / Jose (THANKS!)

DQM viewer layout – SD (all energies)



Data Quality Monitoring (DQM) graphical user interface

Log(E/eV) range: All energies

REF: From 2015/10/04 to 2015/12/17

DQM: From 2016/10/01 to 2016/12/31

$\mathrm{Log}_{10}(E/eV)$ - Log(E/eV) = [All energies]

μ: 0.617

σ: 0.278

χ^2: 73.841

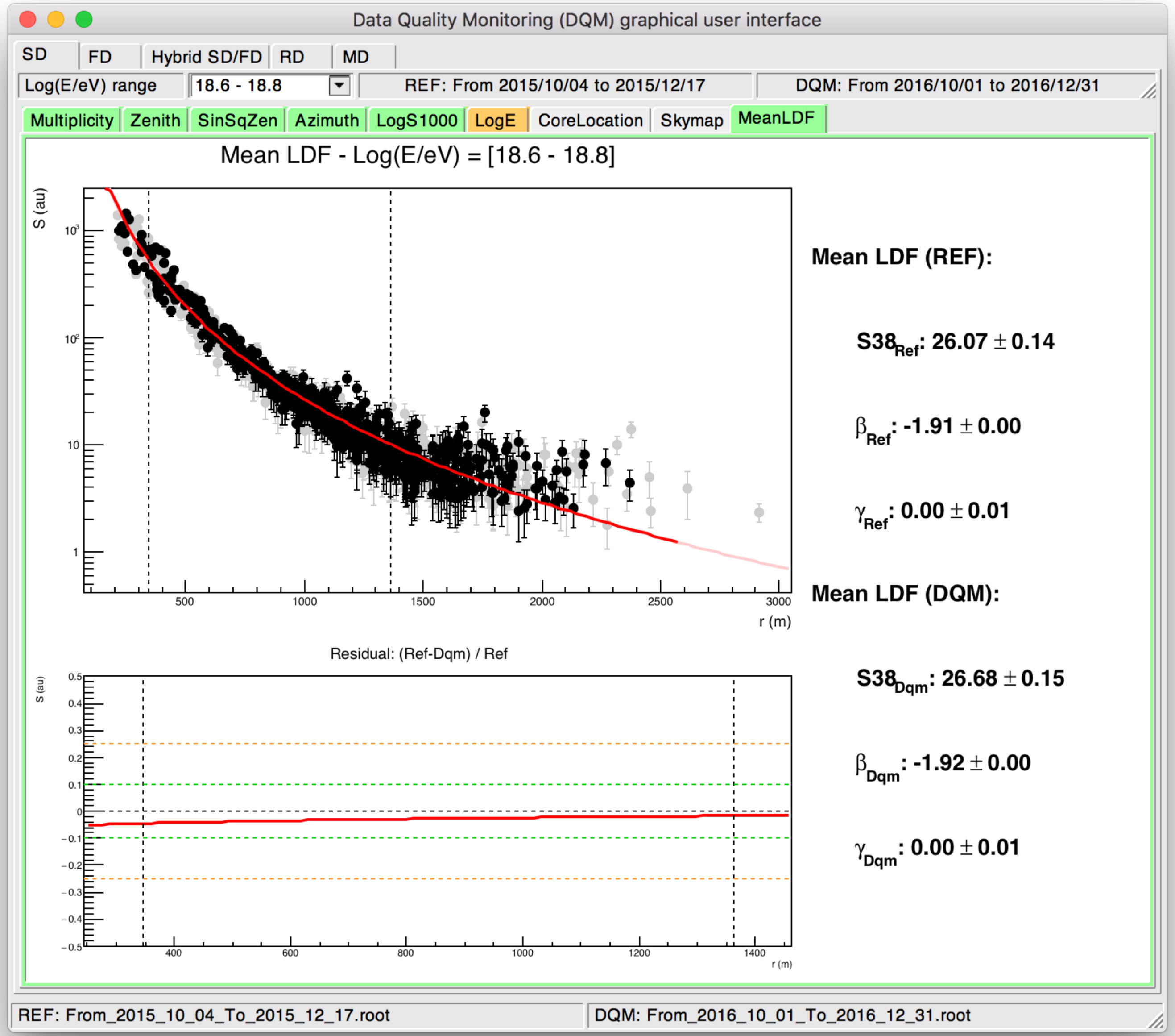
NDoF: 62

χ^2/NDoF: 1.191

REF: From_2015_10_04_To_2015_12_17.root

DQM: From_2016_10_01_To_2016_12_31.root

DQM viewer layout – SD (LDF log(E/eV)=18.6 to 18.8)



Data Quality Monitoring (DQM) graphical user interface

Log(E/eV) range: 18.6 - 18.8

REF: From 2015/10/04 to 2015/12/17

DQM: From 2016/10/01 to 2016/12/31

Mean LDF (REF):

$S38_{Ref}$: 26.07 ± 0.14

β_{Ref}: -1.91 ± 0.00

γ_{Ref}: 0.00 ± 0.01

Mean LDF (DQM):

$S38_{Dqm}$: 26.68 ± 0.15

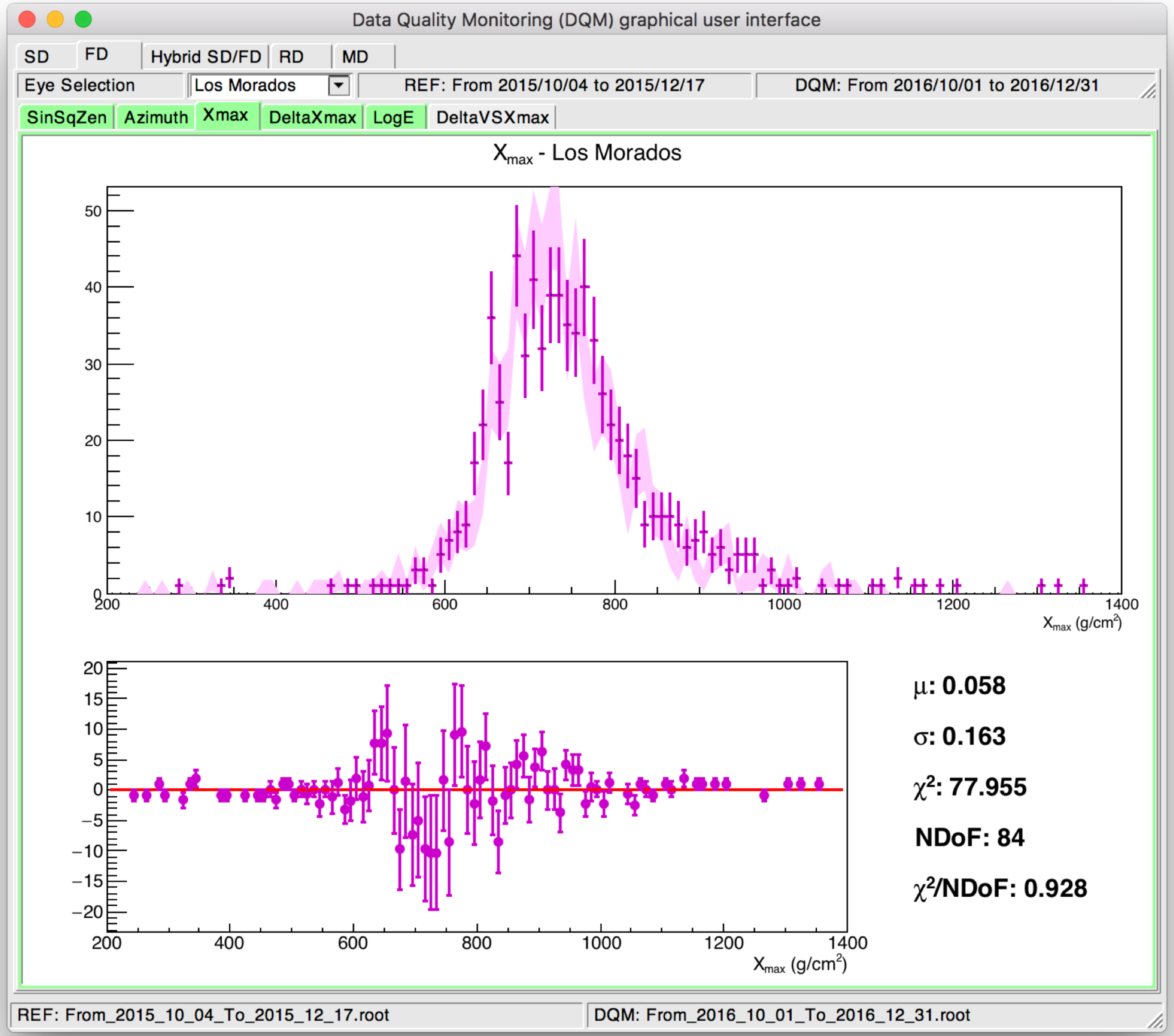
β_{Dqm}: -1.92 ± 0.00

γ_{Dqm}: 0.00 ± 0.01

REF: From_2015_10_04_To_2015_12_17.root

DQM: From_2016_10_01_To_2016_12_31.root

DQM viewer layout – FD (ex: Los Morados)



Eye Selection: Los Morados

REF: From 2015/10/04 to 2015/12/17

DQM: From 2016/10/01 to 2016/12/31

μ: 0.058

σ: 0.163

χ^2: 77.955

NDoF: 84

χ^2/NDoF: 0.928

REF: From_2015_10_04_To_2015_12_17.root

DQM: From_2016_10_01_To_2016_12_31.root

DQMviewer_v1

FRAMEWORK:

- Selected histograms / graphs are extracted from the ADSTs separately
- Stored in files, which are called as argument of the DQM viewer
- Pros:
    - Easy to use
    - Histograms are created outside of the DQM viewer framework
- Cons:
    - Two programs to maintain
    - Cuts need to be handled by histogram creator, not in DQM viewer

DQMviewer_v1

STRUCTURE

- DQMviewer_v1.tar.gz
  - README.txt
  - Two folders
    - DQMadst – To create the period files from ADST

      “make” & done

      - CreatePeriodFile.cc
      - Makefile
      - Three folders with test period root files for testing
    - DQMgui – To run the DQM viewer

      Create dictionary
then
“make” & done

      - DQMviewer.cc
      - DQMviewer.h
      - LinkDef.h
      - Makefile

DQMviewer_v1

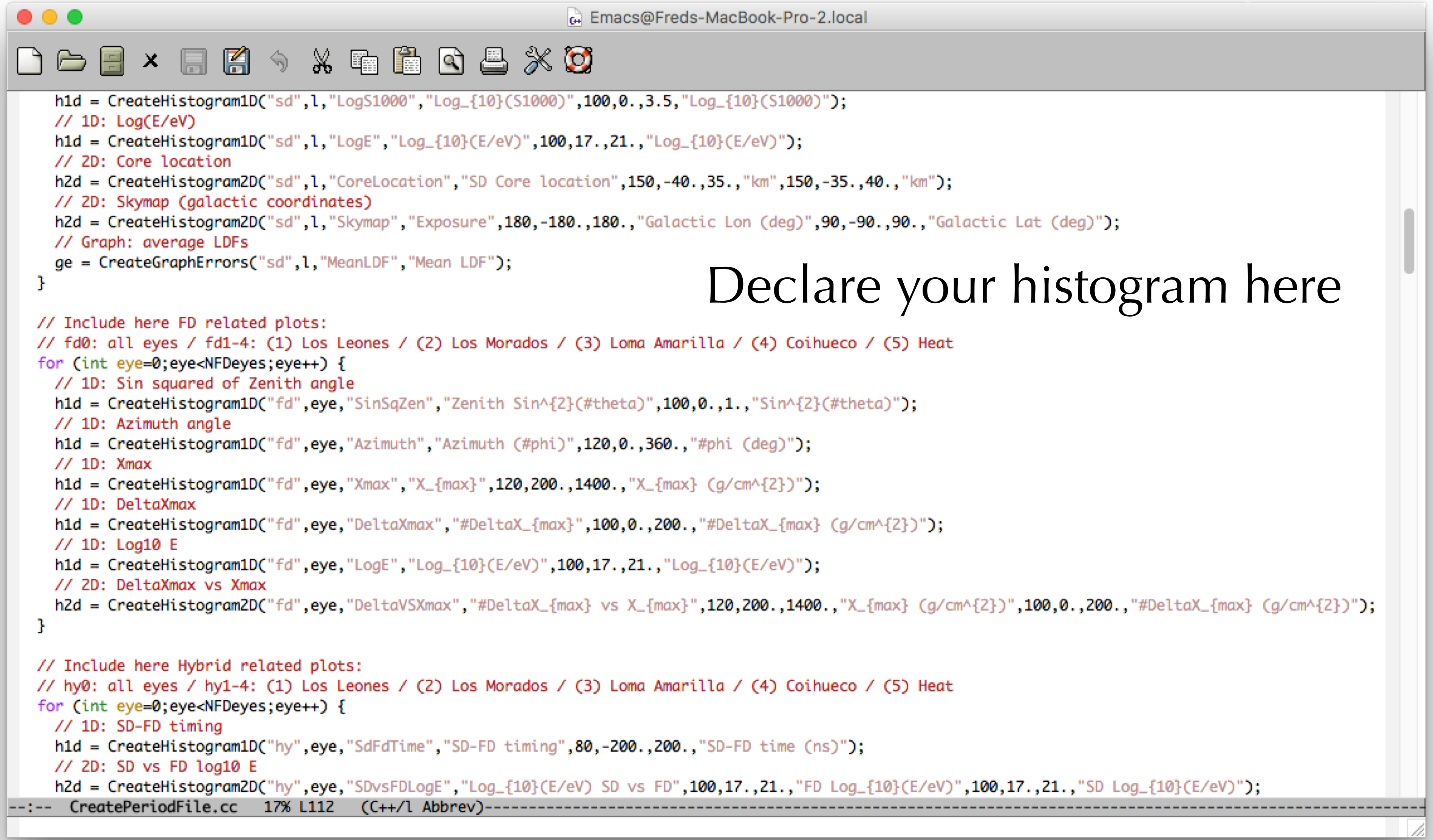
HOW TO RUN

- In DQMadst folder:
  - ./CreatePeriodFile <ADSTfiles.root>
  - Saved in: From_<year>_<month>_<day>_To _<year>_<month>_<day>.root

- In DQMgui folder:
  - ./DQMviewer <REFperiod.root> <DQMperiod.root> *(*)*
  - Enjoy!

(*) using the format: From_< year>_<month>_<day>_To _<year>_<month>_<day>.root

CreatePeriodFile.cc

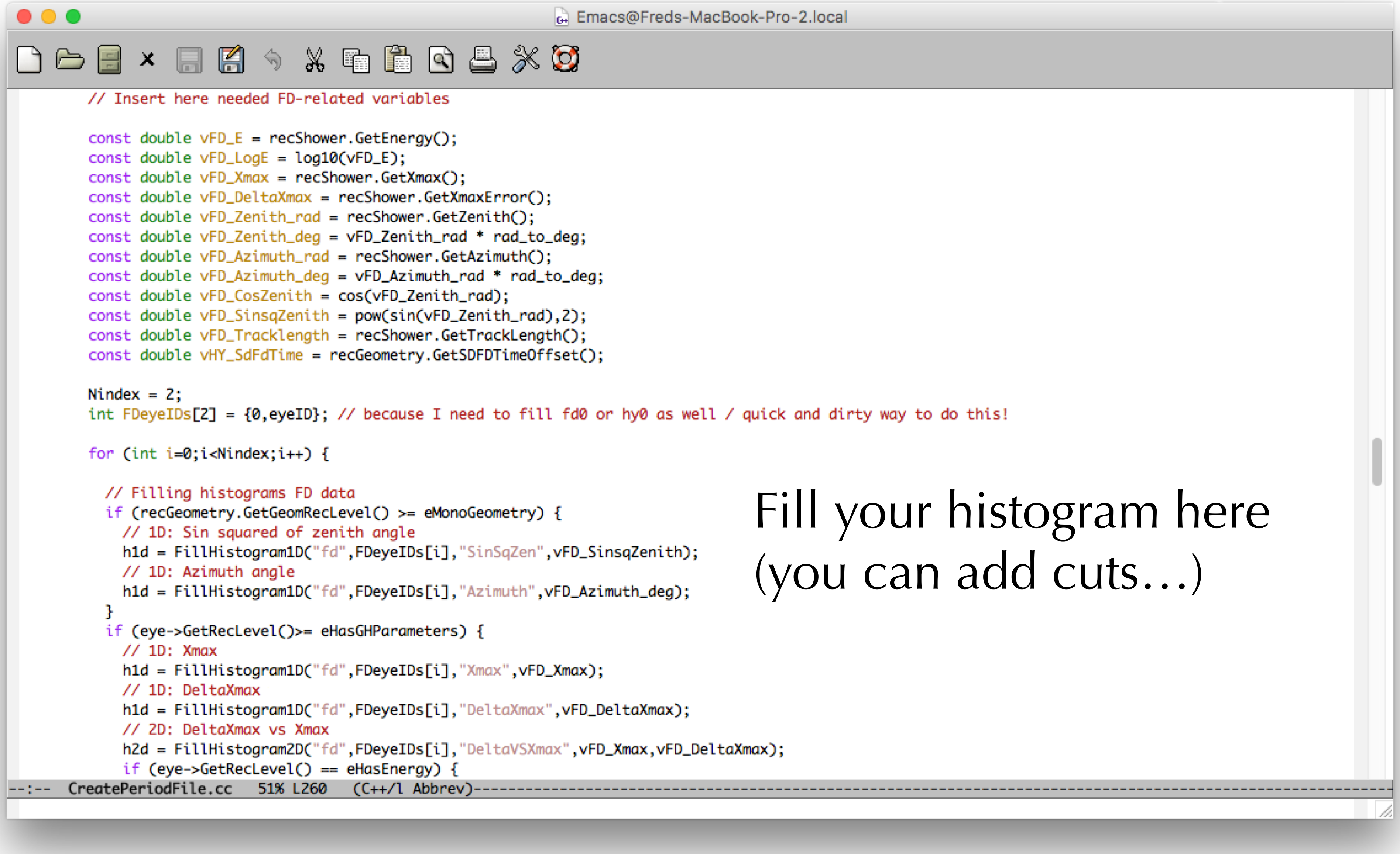


```
    h1d = CreateHistogram1D("sd",l,"LogS1000","Log_{10}(S1000)",100,0.,3.5,"Log_{10}(S1000)");
    // 1D: Log(E/eV)
    h1d = CreateHistogram1D("sd",l,"LogE","Log_{10}(E/eV)",100,17.,21.,"Log_{10}(E/eV)");
    // 2D: Core location
    h2d = CreateHistogram2D("sd",l,"CoreLocation","SD Core location",150,-40.,35.,"km",150,-35.,40.,"km");
    // 2D: Skymap (galactic coordinates)
    h2d = CreateHistogram2D("sd",l,"Skymap","Exposure",180,-180.,180.,"Galactic Lon (deg)",90,-90.,90.,"Galactic Lat (deg)");
    // Graph: average LDFs
    ge = CreateGraphErrors("sd",l,"MeanLDF","Mean LDF");
  }

  // Include here FD related plots:
  // fd0: all eyes / fd1-4: (1) Los Leones / (2) Los Morados / (3) Loma Amarilla / (4) Coihueco / (5) Heat
  for (int eye=0;eye<NFDeyes;eye++) {
    // 1D: Sin squared of Zenith angle
    h1d = CreateHistogram1D("fd",eye,"SinSqZen","Zenith Sin^{2}(#theta)",100,0.,1.,"Sin^{2}(#theta)");
    // 1D: Azimuth angle
    h1d = CreateHistogram1D("fd",eye,"Azimuth","Azimuth (#phi)",120,0.,360.,"#phi (deg)");
    // 1D: Xmax
    h1d = CreateHistogram1D("fd",eye,"Xmax","X_{max}",120,200.,1400.,"X_{max} (g/cm^{2})");
    // 1D: DeltaXmax
    h1d = CreateHistogram1D("fd",eye,"DeltaXmax","#DeltaX_{max}",100,0.,200.,"#DeltaX_{max} (g/cm^{2})");
    // 1D: Log10 E
    h1d = CreateHistogram1D("fd",eye,"LogE","Log_{10}(E/eV)",100,17.,21.,"Log_{10}(E/eV)");
    // 2D: DeltaXmax vs Xmax
    h2d = CreateHistogram2D("fd",eye,"DeltaVSXmax","#DeltaX_{max} vs X_{max}",120,200.,1400.,"X_{max} (g/cm^{2})",100,0.,200.,"#DeltaX_{max} (g/cm^{2})");
  }

  // Include here Hybrid related plots:
  // hy0: all eyes / hy1-4: (1) Los Leones / (2) Los Morados / (3) Loma Amarilla / (4) Coihueco / (5) Heat
  for (int eye=0;eye<NFDeyes;eye++) {
    // 1D: SD-FD timing
    h1d = CreateHistogram1D("hy",eye,"SdFdTime","SD-FD timing",80,-200.,200.,"SD-FD time (ns)");
    // 2D: SD vs FD log10 E
    h2d = CreateHistogram2D("hy",eye,"SDvsFDLogE","Log_{10}(E/eV) SD vs FD",100,17.,21.,"FD Log_{10}(E/eV)",100,17.,21.,"SD Log_{10}(E/eV)");
```

```
--:--  CreatePeriodFile.cc    17% L112    (C++/l Abbrev)----------------------------------------------------------------
```

Declare your histogram here

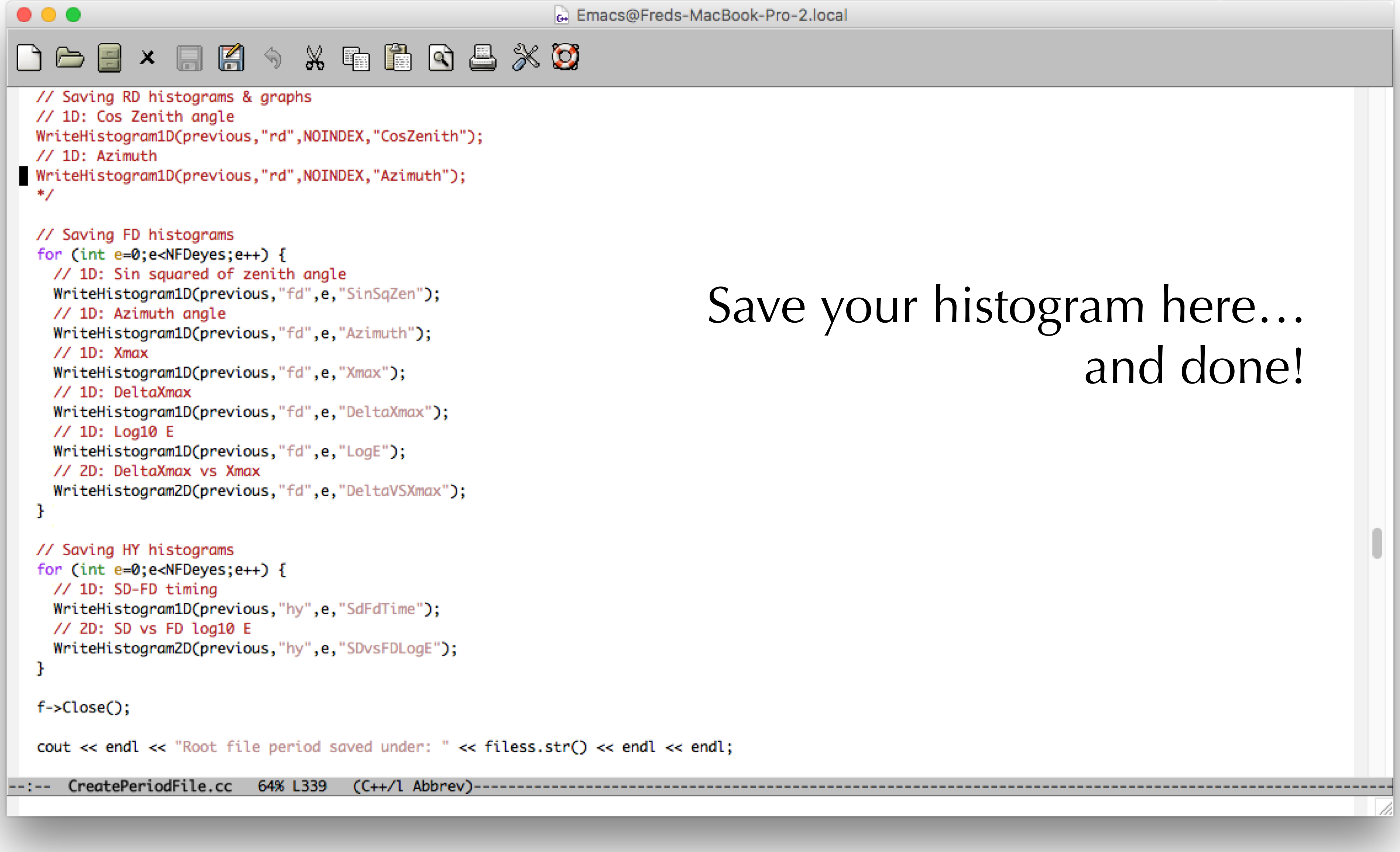
CreatePeriodFile.cc



```
    // Insert here needed FD-related variables

    const double vFD_E = recShower.GetEnergy();
    const double vFD_LogE = log10(vFD_E);
    const double vFD_Xmax = recShower.GetXmax();
    const double vFD_DeltaXmax = recShower.GetXmaxError();
    const double vFD_Zenith_rad = recShower.GetZenith();
    const double vFD_Zenith_deg = vFD_Zenith_rad * rad_to_deg;
    const double vFD_Azimuth_rad = recShower.GetAzimuth();
    const double vFD_Azimuth_deg = vFD_Azimuth_rad * rad_to_deg;
    const double vFD_CosZenith = cos(vFD_Zenith_rad);
    const double vFD_SinsqZenith = pow(sin(vFD_Zenith_rad),2);
    const double vFD_Tracklength = recShower.GetTrackLength();
    const double vHY_SdFdTime = recGeometry.GetSDFDTimeOffset();

    Nindex = 2;
    int FDeyeIDs[2] = {0,eyeID}; // because I need to fill fd0 or hy0 as well / quick and dirty way to do this!

    for (int i=0;i<Nindex;i++) {

      // Filling histograms FD data
      if (recGeometry.GetGeomRecLevel() >= eMonoGeometry) {
        // 1D: Sin squared of zenith angle
        h1d = FillHistogram1D("fd",FDeyeIDs[i],"SinSqZen",vFD_SinsqZenith);
        // 1D: Azimuth angle
        h1d = FillHistogram1D("fd",FDeyeIDs[i],"Azimuth",vFD_Azimuth_deg);
      }
      if (eye->GetRecLevel()>= eHasGHParameters) {
        // 1D: Xmax
        h1d = FillHistogram1D("fd",FDeyeIDs[i],"Xmax",vFD_Xmax);
        // 1D: DeltaXmax
        h1d = FillHistogram1D("fd",FDeyeIDs[i],"DeltaXmax",vFD_DeltaXmax);
        // 2D: DeltaXmax vs Xmax
        h2d = FillHistogram2D("fd",FDeyeIDs[i],"DeltaVSXmax",vFD_Xmax,vFD_DeltaXmax);
        if (eye->GetRecLevel() == eHasEnergy) {
--:--  CreatePeriodFile.cc    51% L260    (C++/l Abbrev)------------------------------------------------------------------------
```

Fill your histogram here
(you can add cuts…)

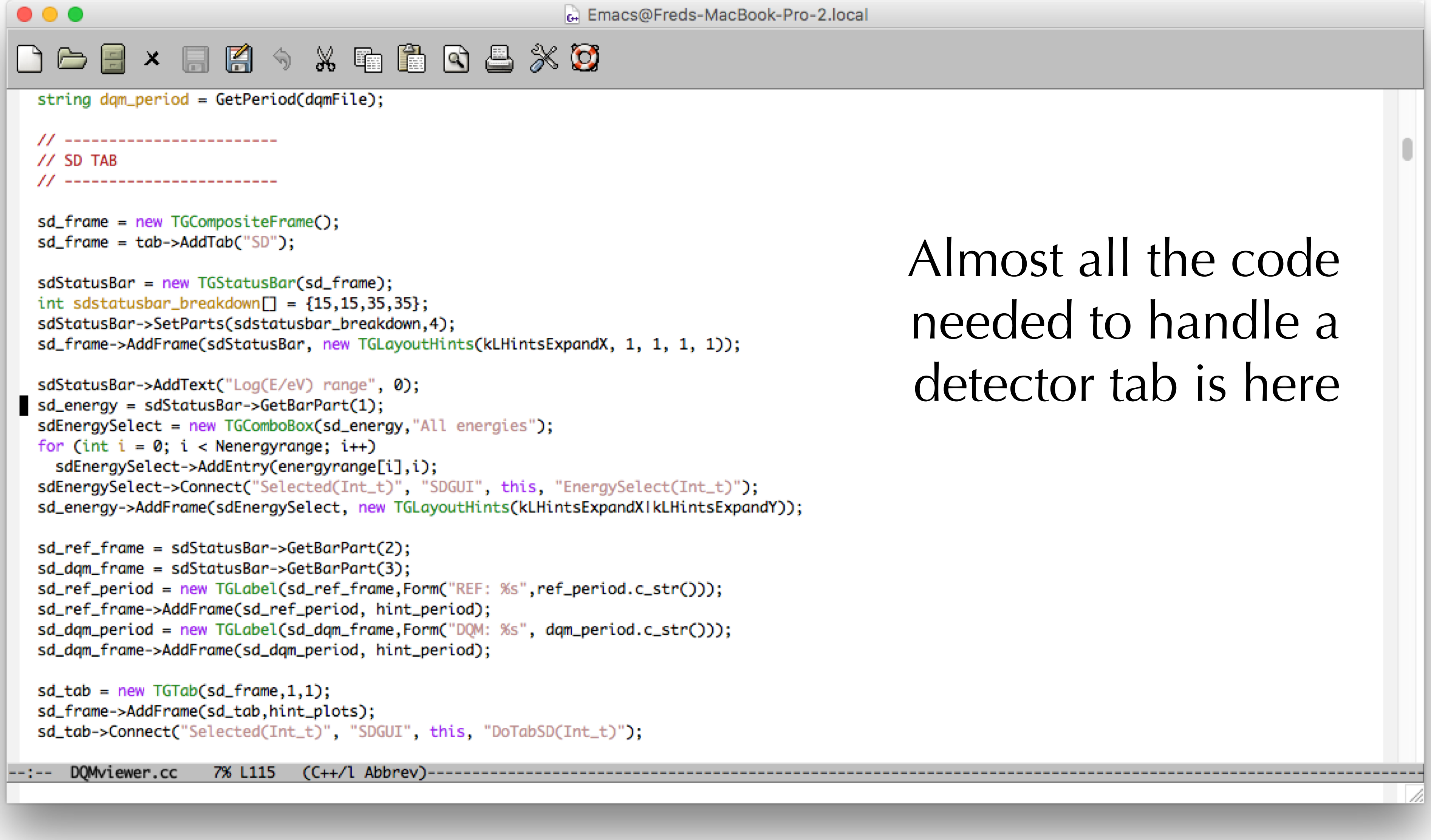
CreatePeriodFile.cc


Emacs@Freds-MacBook-Pro-2.local
--:-- CreatePeriodFile.cc   64% L339   (C++/l Abbrev)

```
// Saving RD histograms & graphs
// 1D: Cos Zenith angle
WriteHistogram1D(previous,"rd",NOINDEX,"CosZenith");
// 1D: Azimuth
WriteHistogram1D(previous,"rd",NOINDEX,"Azimuth");
*/

// Saving FD histograms
for (int e=0;e<NFDeyes;e++) {
  // 1D: Sin squared of zenith angle
  WriteHistogram1D(previous,"fd",e,"SinSqZen");
  // 1D: Azimuth angle
  WriteHistogram1D(previous,"fd",e,"Azimuth");
  // 1D: Xmax
  WriteHistogram1D(previous,"fd",e,"Xmax");
  // 1D: DeltaXmax
  WriteHistogram1D(previous,"fd",e,"DeltaXmax");
  // 1D: Log10 E
  WriteHistogram1D(previous,"fd",e,"LogE");
  // 2D: DeltaXmax vs Xmax
  WriteHistogram2D(previous,"fd",e,"DeltaVSXmax");
}

// Saving HY histograms
for (int e=0;e<NFDeyes;e++) {
  // 1D: SD-FD timing
  WriteHistogram1D(previous,"hy",e,"SdFdTime");
  // 2D: SD vs FD log10 E
  WriteHistogram2D(previous,"hy",e,"SDvsFDLogE");
}

f->Close();

cout << endl << "Root file period saved under: " << filess.str() << endl << endl;
```

Save your histogram here…
and done!

DQMviewer.cc


Emacs@Freds-MacBook-Pro-2.local
--:-- DQMviewer.cc   7% L115   (C++/l Abbrev)

```cpp
string dqm_period = GetPeriod(dqmFile);

// ------------------------
// SD TAB
// ------------------------

sd_frame = new TGCompositeFrame();
sd_frame = tab->AddTab("SD");

sdStatusBar = new TGStatusBar(sd_frame);
int sdstatusbar_breakdown[] = {15,15,35,35};
sdStatusBar->SetParts(sdstatusbar_breakdown,4);
sd_frame->AddFrame(sdStatusBar, new TGLayoutHints(kLHintsExpandX, 1, 1, 1, 1));

sdStatusBar->AddText("Log(E/eV) range", 0);
sd_energy = sdStatusBar->GetBarPart(1);
sdEnergySelect = new TGComboBox(sd_energy,"All energies");
for (int i = 0; i < Nenergyrange; i++)
  sdEnergySelect->AddEntry(energyrange[i],i);
sdEnergySelect->Connect("Selected(Int_t)", "SDGUI", this, "EnergySelect(Int_t)");
sd_energy->AddFrame(sdEnergySelect, new TGLayoutHints(kLHintsExpandX|kLHintsExpandY));

sd_ref_frame = sdStatusBar->GetBarPart(2);
sd_dqm_frame = sdStatusBar->GetBarPart(3);
sd_ref_period = new TGLabel(sd_ref_frame,Form("REF: %s",ref_period.c_str()));
sd_ref_frame->AddFrame(sd_ref_period, hint_period);
sd_dqm_period = new TGLabel(sd_dqm_frame,Form("DQM: %s", dqm_period.c_str()));
sd_dqm_frame->AddFrame(sd_dqm_period, hint_period);

sd_tab = new TGTab(sd_frame,1,1);
sd_frame->AddFrame(sd_tab,hint_plots);
sd_tab->Connect("Selected(Int_t)", "SDGUI", this, "DoTabSD(Int_t)");
```

Almost all the code needed to handle a detector tab is here

Summary

- A Data Quality Monitoring (DQM) standalone viewer is being developed
  - Framework is simple to use and has been tested
  - Version 1 essentially ready for release
- Could use some inputs / contributions
  - ADST pre-processing or ADSTs as inputs?
  - Relevant plots to be included – additional tabs (RD, MD…)
  - Improved quality flagging...
- **Lowers the threshold to carry out long-term performance analyses!**
- FOR A DEMO – JUST ASK ME!