



Segundo Taller de Cómputo de la Colaboración MexNICA

Reconstrucción en el marco de trabajo del MpdROOT

Dra. I. Maldonado¹, Dra. I. Domínguez¹, Dra. Ma. E. Tejeda-Yeomans^{2,3}

¹ Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Sinaloa, ² Facultad de Ciencias - CUICBAS, Universidad de Colima, ³ Departamento de Física, Universidad de Sonora

July 2, 2020

Outline

Reconstrucción

Análisis

Macro reco.C

Una vez llevada a cabo la simulación de eventos, usando el generador UrQMD y la simulación del detector con el macro runMC.C, debemos llevar a cabo el proceso de reconstrucción, lo cual lo hacemos con el macro **reco.C**

- ▶ Cargamos mpdroot y copiamos el archivo reco.C a nuestro directorio de trabajo

```
source ~/Software/mpdroot/build/config.sh  
cp ~/Software/mpdroot/macro/mpd/reco.C .
```

- ▶ Revisar que en el archivo mpdloadlibs.C se cambie:
libMCStack → libMpdMCStack

Argumentos para llevar a cabo la reconstrucción:

- ▶ Archivo de entrada de la simulación: **evetest.root**
- ▶ Archivo de salida de la reconstrucción: **mpddst.root**
- ▶ Número de evento a partir del cual se inicia la reconstrucción: por defceto **0**
- ▶ Número de eventos a analizar: **20** en este caso
- ▶ Tipo de análisis: **local**

```
void reco(TString inFile = "evetest.root", TString outFile = "mpddst.root",  
Int_t nStartEvent = 0, Int_t nEvents = 20, TString run_type = "local")
```

Los elementos más relevantes en la reconstrucción son:

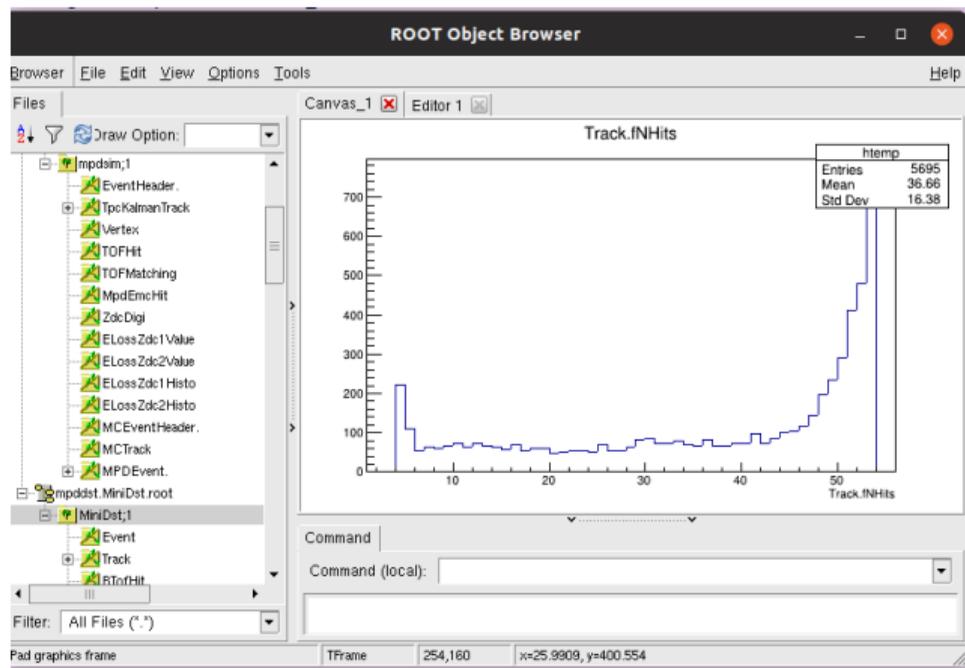
- ▶ FairRunAna* fRun → maneja el proceso de reconstrucción
- ▶ Permite configurar el proceso de reconstrucción

```
MpdKalmanFilter *kalman = MpdKalmanFilter::Instance("KF");  
fRun->AddTask(kalman);
```

```
FairTask* findVtx = new MpdKfPrimaryVertexFinder("Vertex finder");  
fRun->AddTask(findVtx);
```

```
fRun->Init();
```

Archivo mpddst.root y mpddst.Minidst.root



Corrección de DCA

Las trazas reconstruidos contienen información sobre la distancia de máximo acercamiento, el cual requiere una corrección para la selección de las trazas primarias, para ello se tiene que ajustar los valores de la helicidad

- ▶ Copiamos los archivos rootlogon.C y restore_dca.c a nuestro directorio de trabajo

```
cp ~/Software/mpdroot/macro/physical_analysis/Flow/restore_dca/rootlogon.C .
```

```
cp ~/Software/mpdroot/macro/physical_analysis/Flow/restore_dca/restore_dca.c .
```

```
cp ~/Software/mpdroot/macro/physical_analysis/Flow/restore_dca/restore_dca.c .
```

- ▶ Editamos el archivo rootlogon.C

```
"-I${Boost_INCLUDE_DIRS} " =====>>>>> "-I${FAIRROOTPATH}/include ";
```

- ▶ Y realizamos los siguientes pasos

```
root -b  
root [0] gROOT->LoadMacro("../mpdloadlibs.C")  
mpdloadlibs();  
.L restore_dca.c+  
restore_dca("mpddst.root", "mpddst2.root")
```

- ▶ y nos da el archivo con la corrección mpddst2.root

```
lamaldonado@supernova:~/Software/Taller2/correction$ root -b
```

```
-----  
| Welcome to ROOT 6.16/00                https://root.cern |  
|                                     (c) 1995-2018, The ROOT Team |  
| Built for linuxx8664gcc on Feb 24 2020, 16:53:00 |  
| From tags/v6-16-00@v6-16-00 |  
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.'q' |  
-----
```

```
Setting include path ...
```

```
root [0] gROOT->LoadMacro("../mpdloadlibs.C")
```

```
(int) 0
```

```
root [1] mpdloadlibs();
```

```
root [2] .L restore_dca.c+
```

```
root [3] restore_dca("mpdst.root", "mpdst2.root")
```

```
EVENT N 0
```

```
EVENT N 1
```

```
EVENT N 2
```

```
EVENT N 3
```

```
EVENT N 4
```

```
EVENT N 5
```

```
EVENT N 6
```

```
EVENT N 7
```

```
EVENT N 8
```

```
EVENT N 9
```

```
EVENT N 10
```

```
EVENT N 11
```

```
EVENT N 12
```

```
EVENT N 13
```

```
EVENT N 14
```

```
EVENT N 15
```

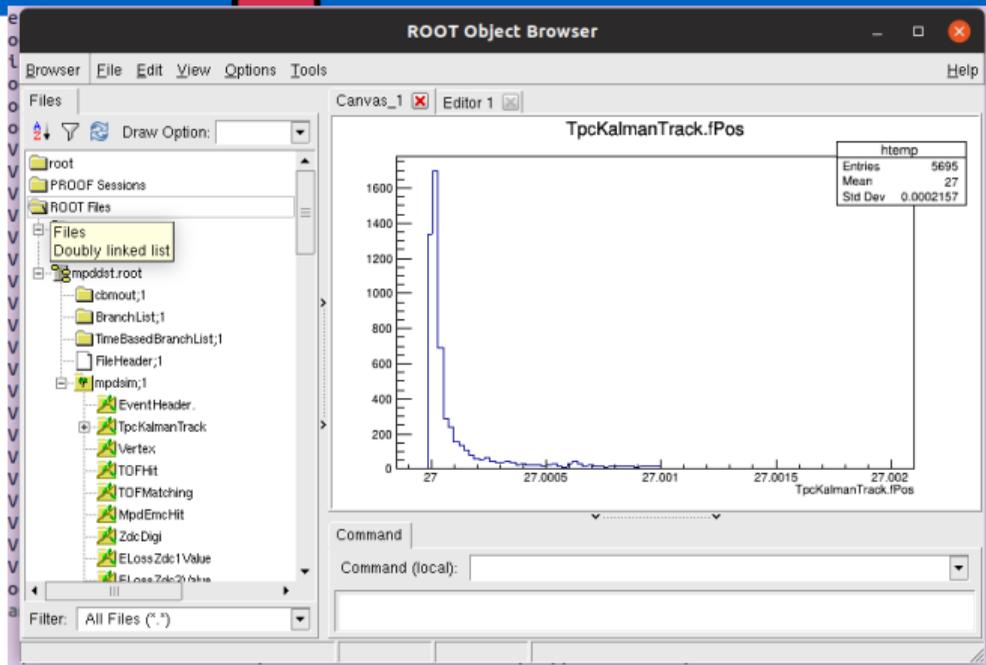
```
EVENT N 16
```

```
EVENT N 17
```

```
EVENT N 18
```

```
EVENT N 19
```

```
root [4] █
```



Más archivos de calibración para el análisis de Flujo, están descritos en:

[https:](https://git.jinr.ru/nica/mpdroot/tree/dev/macro/physical_analysis/Flow)

[//git.jinr.ru/nica/mpdroot/tree/dev/macro/physical_analysis/Flow](https://git.jinr.ru/nica/mpdroot/tree/dev/macro/physical_analysis/Flow)

Macro de análisis

Los diferentes grupos de análisis han desarrollado macros, los cuales se encuentran en:

```
ls ~/Software/mpdroot/macro/physical_analysis/
```

El macro básico es **readDST.C**, que lee el archivo y cuenta el número de trazas por evento.

Este macro podemos modificarlo e incluir las demás ramas del análisis. O bien añadir un histograma.

Ejemplo readDST2.C

Añadimos las siguientes líneas

```
Float_t ptpart = 0;  
TH1F *h1 = new TH1F("h1", "h1-ejemplo", 50, 0, 10);
```

y dentro del loop, llenar el histograma

```
h1->Fill(ptpart);
```

comentar la penultima línea:

```
exit(0);
```

Siempre podemos revisar las clases: *.h y *.cxx, para ver como se llaman a las variables.

Tambien podemos almacenar la salida en un archivo, añadiendo al inicio y al final del archivo

```
TFile out("TestDST2.root","recreate");  ==>>> al inicio
```

```
out.Write();          =====>>> al final  
out.Close();
```

Hay más ejemplos como:

Anal_L0_best.C

Sin embargo no están actualizados y se deben modificar antes de poder correrlos en esta versión, por ejemplo

FairMCTrack → MpdMCTrack

Ahora es tu turno de modificar los archivos