



# 3<sup>er</sup> Taller de Cómputo de la Colaboración MexNICA

Análisis de datos reconstruidos en el marco de trabajo del MpdROOT

Dra. I. Maldonado<sup>1</sup>

ivonne.alicia.maldonado@gmail.com

<sup>1</sup> Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Sinaloa

February 3, 2021

# Outline

Material

Análisis básico

Modificando el macro

# Material

- ▶ Archivo mpddst.root (puede o no estar corregido)
- ▶ macro readDST.C

En caso de no tener los datos reconstruidos, puedes descargarlos de la página del taller:

<https://indico.nucleares.unam.mx/event/1659/session/4/contribution/9/material/0/0.root>

## Macro de análisis

Los diferentes grupos de análisis del experimento MPD:

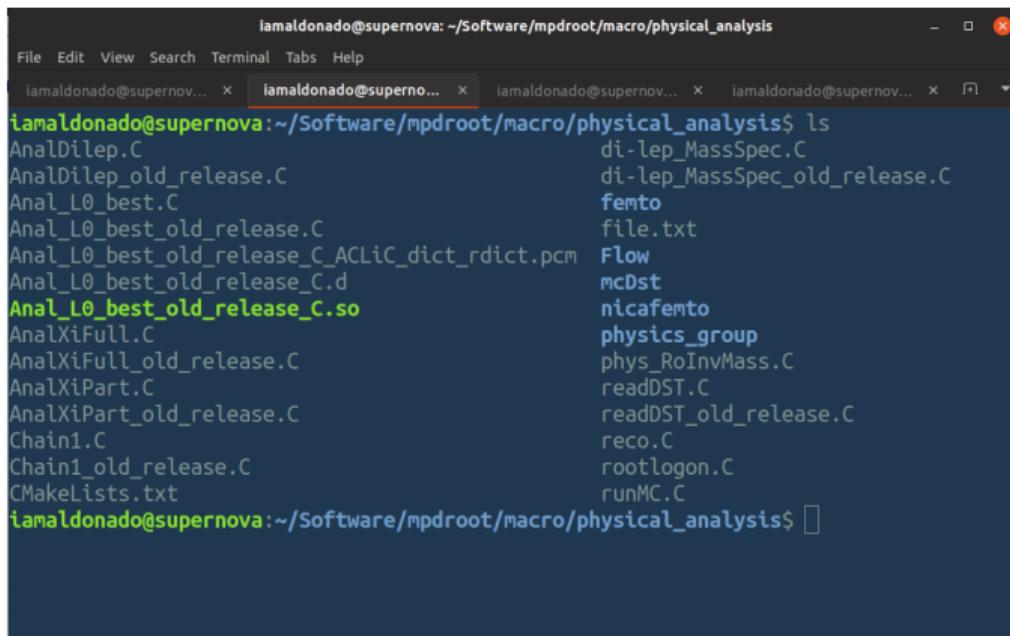
- ▶ PWG1 → Global Observables
- ▶ PWG2 → Spectra and Hypernuclei
- ▶ PWG3 → Correlations and Fluctuations
- ▶ PWG4 → Electromagnetic Probes

han desarrollado diferentes macros, los cuales se encuentran principalmente en:

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

# Macro de análisis

Figure: `ls ~ /mpdroot/macro/physical_analysis`



```
iamaldonado@supernova: ~/Software/mpdroot/macro/physical_analysis
File Edit View Search Terminal Tabs Help
iamaldonado@supernova: ~/Software/mpdroot/macro/physical_analysis$ ls
AnalDilep.C                               di-lep_MassSpec.C
AnalDilep_old_release.C                   di-lep_MassSpec_old_release.C
Anal_L0_best.C                             femto
Anal_L0_best_old_release.C                 file.txt
Anal_L0_best_old_release_C_ACLiC_dict_rdict.pcm Flow
Anal_L0_best_old_release_C.d               mCDst
Anal_L0_best_old_release_C.so              nicafemto
AnalXiFull.C                               physics_group
AnalXiFull_old_release.C                   phys_RoInvMass.C
AnalXiPart.C                               readDST.C
AnalXiPart_old_release.C                   readDST_old_release.C
Chain1.C                                   reco.C
Chain1_old_release.C                       rootlogon.C
CMakeLists.txt                             runMC.C
iamaldonado@supernova: ~/Software/mpdroot/macro/physical_analysis$
```

# Macro readDST.C

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

Antes del algoritmo principal, se definen las clases a usar y se cargan las librerías de mpdroot.

```
es  Vim
readDST.C (-/Software/2doTallerMexNICA/3erTaller/analysis) - GVIM1
File Edit Tools Syntax Buffers Window Help
/* Macro reads DST file produced by macro reco.C */
#include <Rtypes.h>
#include <TChain.h>
#include <TClonesArray.h>
#include <TStopwatch.h>
} Clases a usar

R_ADD_INCLUDE_PATH($VMCWORKDIR)
#include "macro/mpd/mpdloadlibs.C" → Librerías

void readDST(TString in = "") {
  if (in.IsNull()) {
    cout << "Please, provide an input DST-file!" << endl;
    return;
  }

  TStopwatch timer;
  timer.Start();

  TChain *dstTree = new TChain("mpdsim");
  dstTree->Add(in.Data());

  // Activate branches
  MpdEvent *event = nullptr;
  dstTree->SetBranchAddresses("MPDEvent.", &event);
  TClonesArray *fMCTracks = nullptr;
  dstTree->SetBranchAddresses("fMCTracks", &fMCTracks);

  Int_t nEvents = dstTree->GetEntries();
  cout << "Number of events in DST file = " << nEvents << endl;

  for (Int_t i = 0; i < nEvents; i++) {
    dstTree->GetEntry(i);

    Int_t nTracks = event->GetGlobalTracks()->GetEntriesFast();

    cout << "Number of tracks = " << nTracks << endl;
    for (Int_t iTrack = 0; iTrack < nTracks; iTrack++) {
      MpdTrack* track = (MpdTrack*) event->GetGlobalTracks()->UncheckedAt(iTrack);
    } // track loop
  } // event loop

  timer.Print();

  cout << "Test passed" << endl;
  cout << "All ok" << endl;
  exit(0);
}

"readDST.C" 48L, 1288C written
48,1 All
```

Algoritmo principal

## Corriendo el macro.

Introducimos el nombre del archivo a analizar en la siguiente línea

```
void readDST(TSring in = "" )
```

por

```
void readDST(TSring in = "corrected.root" )
```

corremos el macro con

```
$ root readDST.C
```

# ¿Qué información lee?

```
#include <Rtypes.h>
#include <TChain.h>
#include <TClonesArray.h>
#include <TStopwatch.h>

R_ADD_INCLUDE_PATH($VMCWORKDIR)
#include "macro/mpd/mpdloadlibs.C"

void readDST(TString in = "") {
  if (in.IsNull()) {
    cout << "Please, provide an input DST-file!" << endl;
    return;
  }

  TStopwatch timer;
  timer.Start();

  TChain *dstTree = new TChain("mpdsim");
  dstTree->Add(in.Data());

  // Activate branches
  Mpdevent *event = nullptr;
  dstTree->SetBranchAddresses("MPDEvent.", &event);
  TClonesArray *fNCTracks = nullptr;
  dstTree->SetBranchAddresses("MCTrack", &fNCTracks);

  Int_t events = dstTree->GetEntries();
  cout << " Number of events in DST file = " << events << endl;

  for (Int_t i = 0; i < events; i++) {
    dstTree->GetEntry(i);

    Int_t Ntracks = event->GetGlobalTracks()->GetEntriesFast();
    cout << " Number of tracks = " << Ntracks << endl;
    for (Int_t iTrack = 0; iTrack < Ntracks; iTrack++) {
      Mpdevent* track = (Mpdevent*) event->GetGlobalTracks()->UncheckedAt(iTrack);
    } // track loop
  } // event loop
}
```

El nombre del árbol, **mpdsim** y las ramas habilitadas en el macro reco.C:  
**MPDEvent**, **MCTrack**

# ¿Qué información lee?

Ciclo sobre eventos

```
// Activate branches
MpdEvent *event = nullptr;
dstTree->SetBranchAddress("MPDEvent.", &event);
TClonesArray *fMCTracks = nullptr;
dstTree->SetBranchAddress("MCTrack", &fMCTracks);

Int_t events = dstTree->GetEntries();
cout << " Number of events in DST file = " << events << endl;

for (Int_t i = 0; i < events; i++) {
  dstTree->GetEntry(i);

  Int_t Ntracks = event->GetGlobalTracks()->GetEntriesFast();

  cout << " Number of tracks = " << Ntracks << endl;
  for (Int_t iTrack = 0; iTrack < Ntracks; iTrack++) {
    MpdTrack* track = (MpdTrack*) event->GetGlobalTracks()->UncheckedAt(iTrack)
  } // track loop
} // event loop

timer.Print();

cout << " Test passed" << endl;
cout << " All ok " << endl;
exit(0);
```

Llama a cada evento

Ciclo sobre MCTracks para cada evento

Llama a cada traza por evento

"readDST.C" 48L, 1288C written 48,1 All  
Slide 7 of 7 Default 6.36/-0.64 0

# ¿Qué información nos da?

```
readDST.C (-/Software/zdoTallerMexNICA/3erTaller/analysis) - GVIM1
/* Macro reads DST file produced by macro reco.C */
#include <Rtypes.h>
#include <TChain.h>
#include <TClonesArray.h>
#include <TStopwatch.h>

R_ADD_INCLUDE_PATH($VMCWORKDIR)
#include "macro/epd/mpdloadlibs.C"

void readDST(TString in = "") {
  if (in.IsNull()) {
    cout << "Please, provide an input DST-file!" << endl;
    return;
  }

  TStopwatch timer;
  timer.Start();

  TChain *dstTree = new TChain("mpdIn");
  dstTree->Add(in.Data());

  // Activate branches
  MpdEvent *event = nullptr;
  dstTree->SetBranchAddresses("MPDEvent.", &event);
  TClonesArray *fMCTracks = nullptr;
  dstTree->SetBranchAddresses("MCTracks", &fMCTracks);

  Int_t nEvents = dstTree->GetEntries();
  cout << " Number of events in DST file = " << nEvents << endl;

  for (Int_t i = 0; i < nEvents; i++) {
    dstTree->GetEntry(i);

    Int_t nTracks = event->GetGlobalTracks()->GetEntriesFast();
    cout << " Number of tracks = " << nTracks << endl;
    for (Int_t iTrack = 0; iTrack < nTracks; iTrack++) {
      MpdTrack* track = (MpdTrack*) event->GetGlobalTracks()->UncheckedAt(iTrack);
    } // track loop
  } // event loop

  timer.Print();

  cout << " Test passed" << endl;
  cout << " All ok " << endl;
  exit(0);
}

"readDST.C" 48L, 1288C written
```

```
lmalDONado@supernova: ~/Software/zdoTallerMexNICA/3erTaller/analysis
$ root readDST.C

-----
| 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' |
-----

root [0]
Processing readDST.C...
Number of events in DST file = 10
Number of tracks = 7
Number of tracks = 0
Number of tracks = 1
Number of tracks = 27
Number of tracks = 4
Number of tracks = 19
Number of tracks = 24
Number of tracks = 31
Number of tracks = 5
Number of tracks = 2
Real time 0:00:00, CP time 0.120
Test passed
All ok
lmalDONado@supernova:~/Software/zdoTallerMexNICA/3erTaller/analysis
$
```

**¡Solo el número de trazas!** ¿Cómo uso la información del MpdTrack?

## ¿Qué contiene la clase MpdTrack?

Reviso el archivo MpdTrack.h, con:

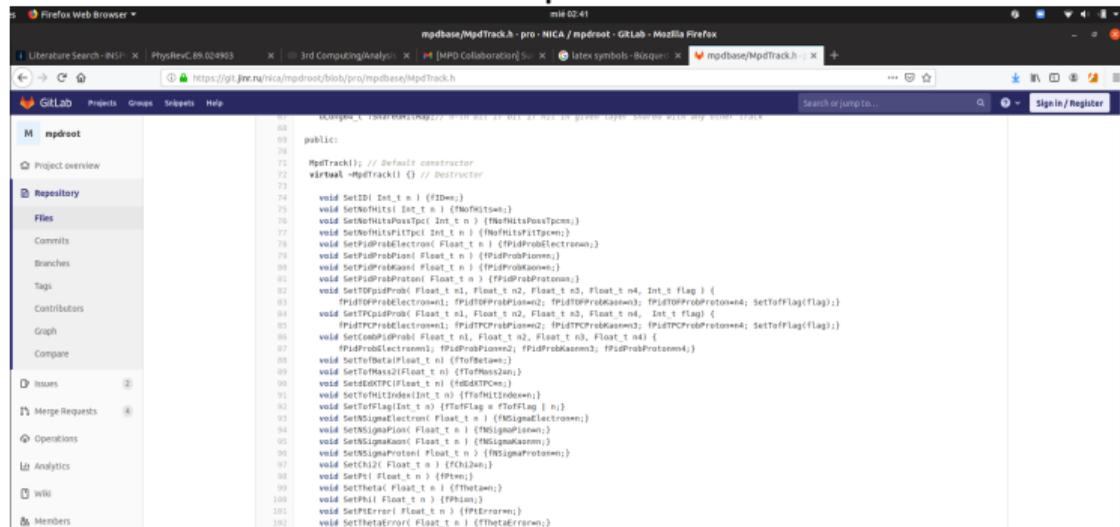
```
less ~/mpdroot/mpdbase/MpdTrack.h
```

o desde el git de la colaboración

<https://git.jinr.ru/nica/mpdroot/blob/pro/mpdbase/MpdTrack.h>

# ¿Qué contiene la clase MpdTrack?

Y revisamos las funciones "public"



```
04 // Copyright © 2018-2022 NICA. All rights reserved. See the LICENSE file for more details.
05
06 public:
07     MpdTrack(); // Default constructor
08     virtual ~MpdTrack() {} // Destructor
09
10     void SetId( Int_t n ) {fID=n;}
11     void SetNoHits( Int_t n ) {fNoHits=n;}
12     void SetNoHitsPosTpc( Int_t n ) {fNoHitsPosTpc=n;}
13     void SetNoHitsPITpc( Int_t n ) {fNoHitsPITpc=n;}
14     void SetProbElectron( Float_t n ) {fProbElectron=n;}
15     void SetProbPion( Float_t n ) {fProbPion=n;}
16     void SetProbKaon( Float_t n ) {fProbKaon=n;}
17     void SetProbProton( Float_t n ) {fProbProton=n;}
18     void SetTfProb( Float_t n1, Float_t n2, Float_t n3, Float_t n4, Int_t flag ) {
19         fTfProbElectron=n1; fTfProbPion=n2; fTfProbKaon=n3; fTfProbProton=n4; SetTfFlag(flag);}
20     void SetTfCpIdProb( Float_t n1, Float_t n2, Float_t n3, Float_t n4, Int_t flag ) {
21         fTfCpIdProbElectron=n1; fTfCpIdProbPion=n2; fTfCpIdProbKaon=n3; fTfCpIdProbProton=n4; SetTfFlag(flag);}
22     void SetSigmaProb( Float_t n1, Float_t n2, Float_t n3, Float_t n4 ) {
23         fSigmaProbElectron=n1; fSigmaProbPion=n2; fSigmaProbKaon=n3; fSigmaProbProton=n4;}
24     void SetTheta( Float_t n ) {fTheta=n;}
25     void SetThetaMax( Float_t n ) {fThetaMax=n;}
26     void SetdGTPC( Float_t n ) {fdGTPC=n;}
27     void SetfMultIndex( Int_t n ) {fMultIndex=n;}
28     void SetfFlag( Int_t n ) {fFlag=n;}
29     void SetSigmaElectron( Float_t n ) {fSigmaElectron=n;}
30     void SetSigmaPion( Float_t n ) {fSigmaPion=n;}
31     void SetSigmaKaon( Float_t n ) {fSigmaKaon=n;}
32     void SetSigmaProton( Float_t n ) {fSigmaProton=n;}
33     void SetChi2( Float_t n ) {fChi2=n;}
34     void SetPhi( Float_t n ) {fPhi=n;}
35     void SetThetaC( Float_t n ) {fThetaC=n;}
36     void SetPhiC( Float_t n ) {fPhiC=n;}
37     void SetPTErr( Float_t n ) {fPErr=n;}
38     void SetThetaErr( Float_t n ) {fThetaErr=n;}
39 }
```

¿Qué variables además del momento transverso de la traza **GetPt()** están definidas?

## ¡Creando un nuevo macro!

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

- ▶ Copiamos el macro readDST.C a readDST2.C
- ▶ Cambiamos el nombre de la función del algoritmo principal readDST → readDST2
- ▶ Y añadir la definición de las variables. En el siguiente ejemplo obtendremos un histograma del momento transverso.

## 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

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

al final del loop sobre eventos, mostrar el histograma

```
h1->Draw();
```

comentar la penultima línea:

```
// exit(0);
```

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();
```

## Reto

Grafica un histograma en 2 dimensiones de la posición del vértice primario.  
Hint: Busca las clases TH2F y MpdVertex

## Otros análisis

Hay más ejemplos como:

- ▶ Anal\_L0\_best.C

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

FairMCTrack → MpdMCTrack

- ▶ Análisis de flujo

[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)

**¡¡Manos a la Obra!!**