CLAS12TOOL is now CLAS12ROOT

Derek Glazier

13/11/19



Clas12root Download

You will first need hipo

git clone --recurse-submodules https://github.com/gavalian/hipo

git clone --recurse-submodules https://github.com/jeffersonlab/clas12root.git

cd clas12root

git checkout tutorial

seteny CLAS12ROOT \$PWD

setenv PATH "\$PATH": "\$CLAS12ROOT/bin"

setenv HIPO /Where/did/I/put/my/hipo

./installC12Root

Get file from ifarm : ~devita/skim.hipo Or http://nuclear.gla.ac.uk/~dglazier/skim.hipo



CLAS12ROOT on ifarm

source /group/clas12/packages/setup.csh (or setup.sh for bash)

module load clas12/pro



Why CLAS12ROOT ?

ROOT provides many useful tools for data analysis CLAS12 HiPO file is compact

```
npcglazier2.dglazier> ls -lh /work/jlab/clas12data/v16/skim9_5038.hipo
-rw-r--r-- 1 dglazier dglazier 3.7G Aug 30 10:25 /work/jlab/clas12data/v16/skim9_5038.hipo
npcglazier2.dglazier> ls dst2root/skim9_5038.hipo.root -lh
-rw-r--r-- 1 dglazier dglazier 5.7G Oct 29 16:40 dst2root/skim9_5038.hipo.root
```

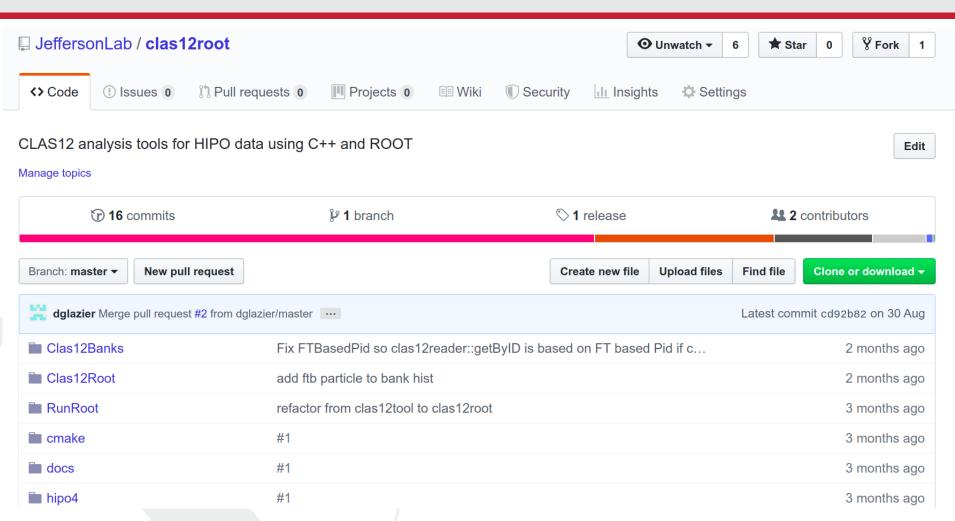
and fast to read (7.5 Gb ROOT 7.0 Gb Hipo SSD)

ROOT	ROOT	HIPO	HIPO
TTREE	DATAFRAME	CLAS12ROOT	CLAS12ROOT dev
438 s	223 s	40 s	28 s

And is the default format for DSTs, no overhead for converting to ROOT



CLAS12ROOT @JeffersonLab Github





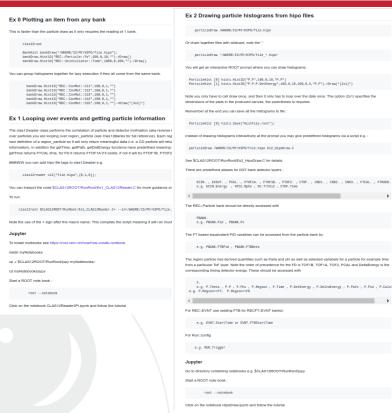
CLAS12ROOT @JeffersonLab Instructions

Clas12Root Data Analysis Tools for bino4 data format Examples are given for running in interactive ROOT sessions and ROOT-Jupyter notebooks Clas12Banks -> Clas12Root NEW We now use an external hipo4 repository. This must be pointed at with the variable HIPO when installing. The files The Class 2 Banks implementation can be used independent of BOOT, although currently BOOT dictionaries are created for Also see c++ function for accessing banks "Cheat sheet" AccesssingBankDataInCpp.txt in the top level directory. clas 12 hipo DSTs To Download To setup Run ROOT seteny CLAS12ROOT \$PWD (the actual path can be added in your bashro or tohro) seteny PATH "\$PATH": "\$CLAS12ROOT/bin" To install ues with cmake and your ROOTSYS you can try using the local FindROOT file. Edit the CMakeList.txt file: removing the lines with comment ##USEROOTSYS and uncomment the line "Note just to belo confuse you the first letter in bank items have been capitilised so where you might expect to use

header files e.g. Clas12Banks4/particle.h and look at the get function declarations e.g. getPz() rather than getpz() **

To start an interactive session with pre-loaded Clas12Root use clas12root instead of root on the command line

interactive root session







CLAS12ROOT @JeffersonLab Notebooks

Creating Root tree files with hiporoot::ParticleTree

First load the classes into the notebook

```
In [ ]: gR00T->ProcessLine(".x $CLAS12R00T/RunRoot/LoadClas12Root.C");

Create the tree maker with the full path to the hipo file you want to analyse. You may also give wildcard (*) arguments .
```

```
In [ ]: ParticleTree treemaker("/WHERE/IS/MY/HIPO/file.hipo","/WHERE/SHOULD/I/PUT/MY/tree.root");
//ParticleTree treemaker("/work/jlab/clas12data/dst_skim4_5038.hipo","tree.root");
```

Create some tree branches. There are predefined aliases for DST bank detector layers :

```
ECIN. , ECOUT. , PCAL. , FT0F1A. , FT0F1B. , FT0F2. , CT0F. , CND1. , CND2. , CND3. , FTCAL. , FTH0D0. , HTCC. , LTCC. , DC. , CVT. e.g. ECIN.Energy , HTCC.Nphe , DC.TrChi2 , CT0F.Time
```

The REC::Particle bank should be directly accessed with

```
e.g. PBANK.Pid , PBANK.Px
```

Ther REC::EVNT bank :

```
EVNT.
e.g. EVNT.StartTime,
```

The region particle should be accessed with

```
P.
e.g. P.Theta , P.P , P.Phi , P.Region , P.Time , P.DetEnergy , P.DeltaEnergy , P.Path , P.Pid , P.CalcMass
```

Create branches

Here we can create tree branches correlating particle and detector information.

I can make a branch for any DST bank item using the aliases above.

I can choose standard ROOT branch type e.g /F for float /I for int /D for double,...

```
In []: treemaker.Branch("PBANK.Px/F");
    treemaker.Branch("PBANK.Py/F");
    treemaker.Branch("PBANK.Pz/F");
    treemaker.Branch("PBANK.Vx/F");
    treemaker.Branch("PBANK.Vy/F");
    treemaker.Branch("PBANK.Vy/F");
    treemaker.Branch("PBANK.Pid/I");
    treemaker.Branch("PIDANK.Pid/I");
    treemaker.Branch("P.Time/F");
    treemaker.Branch("P.Time/F");
```

Drawing histograms with hiporoot::ParticleHist

First load the classes into the notebook

```
In [ ]: gROOT->ProcessLine(".x $CLAS12ROOT/RunRoot/LoadClas12Root.C");
```

Create the histogram maker with the full path to the hipo file you want to analyse. You may also give wildcard (*) arguments

```
In [ ]: ParticleHist hists("/WHERE/IS/MY/HIPO/file.hipo");
//ParticleHist hists("/work/jlab/clas12data/dst_skim4_5038.hipo");
```

Turn on javascript ROOT for interactive histograms

```
In [ ]: %jsroot
```

Draw some histograms. There are predefined aliases for DST bank detector layers :

```
ECIN. , ECOUT. , PCAL. , FT0F1A. , FT0F1B. , FT0F2. , CT0F. , CND1. , CND2. , CND3. , FTCAL. , FTH0D0. , HTCC. , LTCC. , DC. , CVT. e.g. ECIN.Energy , HTCC.Nphe , DC.TrChi2 , CT0F.Time
```

The REC::Particle bank should be directly accessed with

```
PBANK.
e.g. PBANK.Pid , PBANK.Px
```

The region particle should be accessed with

```
P.
e.g. P.Theta , P.P , P.Phi , P.Region , P.Time , P.DetEnergy , P.DeltaEnergy , P.Path , P.Pid , P.CalcMass
```

Drawing hists 1

First draw a 1D histogram of the time difference between FTOF1A and FTOF1B

Second, draw a 2D hist of the time difference versus the particle theta, with colour map.

Plot the 2 histograms side-by-side (2x1)

```
In [ ]: hists.Hist1D("FTOF1B.Time-FTOF2.Time",1000,-5,5,"FTOF1B.Time-FTOF2.Time");
hists.Hist2D("FTOF1B.Time-FTOF2.Time:P.Theta*TMath::RadToDeg()",50,-5,5,50,0,40,"FTOF1B.Time-FTO
F2.Time")->Draw("(2x1)coll");
```

Drawing hists 2

Now draw the θ versus ϕ distributions for different particle types

In []: hists.Hist2D("P.Theta*TMath::RadToDeg():P.Phi*TMath::RadToDeg()",180,0,180,180,-180,180,"P.Pid==



CLAS12ROOT @JeffersonLab Forum



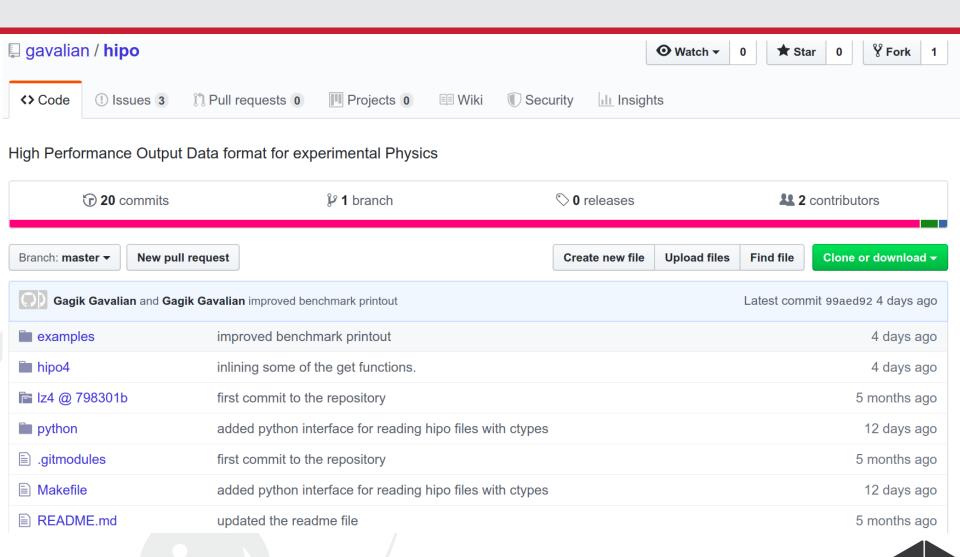
https://clas12.discourse.group/c/clas12root



Clas12Root ▶ all tags ▶ Latest Top			+	New To	pic !
Topic			Replies	Views	Activity
Reading Bank information in clas12root		D 😩	3	20	2d
Same information in CND1 and FTOF1A		R 🛟 😩	4	15	2d
Using FTbased Pid	last visit		0	13	Aug 30
Restructuring Clas12Tool and Hipo		(2) (2)	6	33	Aug 15
Compile a project with g++ and Clas12Tool as a library		G	1	27	Aug 13
Clas12tool on OSX		⊕ B	2	29	Aug 1
Cherenkov # photoelectrons		• •	1	17	Jul 26
★ About the Clas12Root category			0	13	Jul 26



HIPO @gavalian Github



BankHist - drawing items from a bank

Create a new ROOT macro : BankHist.C

```
BankHist bankDraw("/where/is/myHipo.hipo");
bankDraw.Hist1D("REC::Particle::Pid",10000,-3000,3000,"")->Draw();
```

Run:

clas12root BankHist.C



BankHist - drawing items from a bank

Create a new ROOT macro : BankHist.C

```
BankHist bankDraw("/where/is/myHipo.hipo");
bankDraw.Hist2D("REC::Particle::Px:REC::Particle::Py",100,-3,3,100,-3,3,"");
bankDraw.Hist1D("REC::Particle::Pid",10000,-3000,3000,"")->Draw("colz");
}
```

Standard ROOT TH options

Run:

clas12root BankHist.C



BankHist - drawing items from a bank

```
Create a new ROOT macro : BankHist.C
BankHist bankDraw("/where/is/myHipo.hipo");
bankDraw.Hist2D("REC::Particle::Px:REC::Particle::Py",100,-3,3,100,-3,3,"");
bankDraw.Hist1D("REC::Particle::Pid",10000,-3000,3000,"");
bankDraw.Hist1D("sqrt(REC::Particle::Px*REC::Particle::Px"
                "+ REC::Particle::Py*REC::Particle::Py"
                "+ REC::Particle::Pz*REC::Particle::Pz)"
               ,200,0,10,"")->Draw("(3x1)colz");
bankDraw.Save("BankHists.root");
 Run
```

clas12root BankHist.C

ParticleDraw - Drawing particle banks Region particle functions

In addition to actual banks the particle object has some additional functions wehre some choices have been made depending on which region they are in

```
C++ code
                                                   particleDraw
particles[i] \rightarrow getPid();
                                                   P.Pid
particles[i] \rightarrow getTime();
                                                   P.Time
particles[i] \rightarrow getPath();
                                                   P.Path
particles[i] \rightarrow getDeltaEnergy();
                                                   P.DeltaEnergy
particles[i] → getSector();
                                                   P.Sector
                                                                  (FD, CD, FT)
particles[i] \rightarrow getRegion();
                                                   P.Region
particles[i] → getTheta();
                                                   P.Theta
particles[i] \rightarrow getPhi();
                                                   P.Phi
particles[i] \rightarrow getP();
                                                   P.P
```

For FT Time, Path, comes from FTCAL, DeltaEnergy from FTHODO
For FD, Time, Path, DeltaEnergy comes from FTOF1B, FTOF1A, FTOF2, PCAL in order of preference

For CD Time, Path, DeltaEnergy comes from CTOF, then CND if no CTOF

ParticleDraw - Drawing particle banks with Correlated detector banks

Drawing particle e- momentum

particleDraw /where/is/my/hipo/file.hipo

* run from script or type in interactive ROOT command line



ParticleDraw - Drawing particle banks with Correlated detector banks

Drawing particle e- momentum using REC::Particle => PBANK

particleDraw /where/is/my/hipo/file.hipo

```
hists.Hist1D("sqrt(PBANK.Px*PBANK.Px+PBANK.Py*PBANK.Py*PBANK.Pz*PBANK.Pz)",200,0,10,"P.Pid==11")->Draw("");
```

* run from script or type in interactive ROOT command line



ParticleDraw - Drawing particle banks with Time

Bank aliases e.g. FTOF1A, CTOF, CND1, CND2, CND3, FTCAL,

particleDraw /where/is/my/hipo/file.hipo ParticleHist.C

* run from script or type in interactive ROOT command line



ParticleDraw - Drawing particle banks In 2D

Bank aliases e.g. FTOF1A, ECIN , ECOUT, PCAL

particleDraw /where/is/my/hipo/file.hipo ParticleHist2.C



^{*} run from script or type in interactive ROOT command line

Clas12root event loops

This is not a recommended framework for full CLAS12 data analysis

Analysis should be performed in a more general framework which can handle the large data volumes in a reproducable manner

The following just shows how you may use parts of clas12root in such a framework



Clas12root event loops Loop over files

You can do this however you like!

Here I use clas12root::HipoChain normally used for clas12proof

```
HipoChain chain;
chain.Add("/WHERE/IS/MY/HIPO/file.hipo");
```

```
//loop over files
for(int ifile=0;ifile<chain.GetNFiles();++ifile){
  clas12reader c12{chain.GetFileName(ifile).Data()};</pre>
```

* Note, each file gets a new reader



Clas12root event loops create standard ROOT stuff

```
//create particles before looping to be more efficient
TLorentzVector p4_gamma1;
TLorentzVector p4_gamma2;

//Create histogram
TH1F hmass("pi0mass","Invariant Mass to 2#gamma",100,0,0.6);
TH1F htime("DeltaTime","Time difference of 2#gamma",100,-10,10);
```

* You might prefer to use the ROOT GenVector 4-vector, it is faster



Clas12root event loops Configuring the reader

```
Also at least option....
c12.addAtLeastPid(22,2); //at least 2 gamma
```

Clas12root event loops Loop over events and get particles

```
//loop over all events in the file
while(c12.next()==true){
  if(c12.getDetParticles().empty())
    continue:
   auto electron=c12.getByID(11)[0];
   auto gamma1=c12.getByID(22)[0];
   auto gamma2=c12.getByID(22)[1];
   auto proton=c12.getByID(2212)[0];
   auto pip=c12.getByID(211)[0];
   auto pim=c12.getByID(-211)[0];
```



Clas12root event loops Do ROOT stuff



Clas12root other stuff

Filter events into ROOT ntuples particleTree

Run multicore over multiple files
HipoSelector and HipoChain

