

# HiForest

- HiForest – формат данных, содержащий часть информации о событиях.
- Удобство – для работы с ним не нужно использовать CMSSW, только ROOT.
- Содержит информацию о треках.
- Большая часть находится в MIT.
- Структура файлов pPb, pp, PbPb отличается.

# Структура

## Available trees in the HiForest :

- *hiEvtAnalyzer/HiTree* - Event property tree. Contains centrality variables, event plane, vertex position, run number, event number and lumi block information
- *anaTrack (mergedTrack, ppTrack)/trackTree* – Содержит треку.
- *skimanalysis/HltTree* - Skim tree which contains collision event selection, HCAL noise rejection selection, Spike removal selection bool variables.
- *rechitanalyzer/eb* - ECAL Barrel rechit tree
- *rechitanalyzer/hbhe* - HCAL Barrel & Endcap rechit tree
- *rechitanalyzer/hf* - HF rechit tree
- *rechitanalyzer/tower* - CaloTower tree
- *icPu5JetAnalyzer/t* - Jet tree using ICPU5 algorithm. Calo jet with iterative cone subtraction.
- *akPu3PFJetAnalyzer/t* - Jet tree using AKPU3PF algorithm. Particle Flow jet with iterative cone subtraction.
- *pfcandAnalyzer/pfTree* - Particle Flow object tree.
- *anaMET/metTree* - Missing Et tree
- *muonTree/HLTMuTree* - Muon tree, contains global and standalone muons.

# Местонахождение

- Pb-Pb (2.76 TeV):  
/eos/cms/store/group/phys\_heavyions/icali/MinBias\_Merged.root  
( <https://twiki.cern.ch/twiki/bin/viewauth/CMS/HiForest#Data> )
- P-Pb (run 202792):  
/afs/cern.ch/user/t/tuos/work/public/pPb/Data/  
PAPhysics-pt01-full\_WithCentrality.root  
( <https://twiki.cern.ch/twiki/bin/viewauth/CMS/PPbCentralityAna> ,  
<https://twiki.cern.ch/twiki/bin/viewauth/CMS/HiForestPA2013> )
- P-p (mainly run 161438):  
MIT: /store/user/vzhukova/Forest/pp\_merged\_full.root

## ССЫЛКИ

- <https://twiki.cern.ch/twiki/bin/viewauth/CMS/HiForest>
- <https://twiki.cern.ch/twiki/bin/viewauth/CMS/PPbCentralityAna>
- <https://twiki.cern.ch/twiki/bin/viewauth/CMS/HiForestPA2013>
- <https://twiki.cern.ch/twiki/bin/viewauth/CMS/HiForestPA2012>
- [https://ekondrat.web.cern.ch/ekondrat/hiforest\\_read\\_sample.C](https://ekondrat.web.cern.ch/ekondrat/hiforest_read_sample.C)

```
#include <time.h>
#include <iostream>
#include <fstream>
#include <TMath.h>
#include <TString.h>
#include <TTree.h>
#include <TFile.h>
#include <TDirectoryFile.h>

//test
#define _NUMBER_OF_EVENTS 300
//real
#ifndef _NUMBER_OF_EVENTS
#define _NUMBER_OF_EVENTS (nevents-1)
#endif

using namespace std;

void hiforest_read_sample();

int main()
{
    hiforest_read_sample();
}
```

```

void hiforest_read_sample()
{
//Declare the base variables

    int iev; //iterators;
    string current_name;

    Int_t nevents = 0;
    Int_t multiplicity, centrality;
    Float_t phi[40000], pt[40000], eta[40000];

//declare the list of files

    vector<string> name;
    name.push_back("root://eoscms//
eos/cms/store/group/phys_heavyions/icali/MinBias_Merged.root");

//Create the new file

    TFile newfile("../my_data_for_PbPb.root", "recreate");

//Create the output trees in the new file

    TDirectoryFile* anaTrack2 = new TDirectoryFile("anaTrack","anaTrack");
    TTree* tracktree2 = new TTree("trackTree", "trackTree");

    TDirectoryFile* hiev2 = new TDirectoryFile("hiEvtAnalyzer", "hiEvtAnalizer");
    TTree* hitree2 = new TTree("HiTree", "HiTree");

    tracktree2->Branch("nTrk", &multiplicity, "multiplicity/I");
    tracktree2->Branch("trkPhi", phi, "phi[multiplicity]/F");
    tracktree2->Branch("trkPt", pt, "pt[multiplicity]/F");
    tracktree2->Branch("trkEta", eta, "eta[multiplicity]/F");

    hitree2->Branch("hiBin", &centrality, "centrality/I");

```

```

//File cycle; it = name of current file

    for(vector<string>::const_iterator it = name.begin(); it != name.end(); ++it)
    {
        current_name = *it;
//Enter the tree

        TFile *a = TFile::Open(current_name.c_str());

        TDirectoryFile *anaTrack =
(TDirectoryFile*) a -> Get("mergedTrack");

        TTree *tracktree = (TTree*)anaTrack->Get("trackTree");

        TDirectoryFile *hiev =
(TDirectoryFile*) a -> Get("hiEvtAnalyzer");    //for centrality

        TTree *hitree = (TTree*)hiev->Get("HiTree");

//Connect the variables with the branches

        tracktree->SetBranchAddresses("nTrk",&multiplicity);
        tracktree->SetBranchAddresses("trkPhi",&phi);
        tracktree->SetBranchAddresses("trkPt",&pt);
        tracktree->SetBranchAddresses("trkEta",&eta);

        hitree->SetBranchAddresses("hiBin",&centrality);

        nevents = (Int_t)tracktree->GetEntries();           //GetEntries

```

```
//write the information into the trees
```

```
for ( iev = 0; iev < _NUMBER_OF_EVENTS; iev++ )  
{  
    hitree->GetEntry(iev);    //here u've got centrality  
    tracktree->GetEntry(iev); //here u've got multiplicity,phi[],pt[],eta[]  
  
    tracktree2->Fill();  
    hitree2->Fill();  
}  
}
```

```
//Write the necessary data into the newfile
```

```
newfile.cd();  
anaTrack2->cd();  
tracktree2->Write();  
  
hiev2->cd();  
hitree2->Write();  
}
```