Argonne NATIONAL LABORATORY

… for a brighter future

U.S. Department of Energy

UChicago ▶ Argonne LLC

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

Doing ESD analysis at T3g

Belen Salvachua

High Energy Physics Division
Argonne National Laboratory

US ATLAS Analysis Jamboree: ANL

Friday 2nd April 2010

http://indico.cern.ch/conferenceDisplay.py?confId=87268

Outline

- Why do you want to use ESD data for an analysis?
- In case you need to: Where to start?
  - Useful web pages
  - Prepare your area
- Description of the code
- Where is the data?

Why do you want to use ESD data for analysis?

do **NOT use** ESD data unless is totally needed!!!

Reason NOT to use them

AODs usually contain the information that you will need and they are easier to handle:

- Smaller in size

BIG ADVANTAGE!

- They usually require minor database access

Reason to use them

Sometimes you need information that is ONLY on ESD data

Where to start? Useful web pages

- ATLAS computing twiki:
  https://twiki.cern.ch/twiki/bin/view/Atlas/AtlasComputing
- ATLAS Analysis Workbook twiki:
  https://twiki.cern.ch/twiki/bin/view/AtlasProtected/PhysicsAnalysisWorkBookRel15
- GoodRunList twiki:
  https://twiki.cern.ch/twiki//bin/viewauth/Atlas/GoodRunsLists
- AMI database:
  http://ami.in2p3.fr/
- If working at Argonne (T3g Users Guide):
  https://atlaswww.hep.anl.gov/twiki/bin/view/UsAtlasTier3/Tier3gUsersGuide

ATLAS Computing Workbook

Analysis Workbook

ATLAS Computing

Colour Key: *TWiki(unread)* *TWiki(read)* *Old php pages* *External Links*

ATLAS Home

ATLAS TWiki
Detectors
Trigger
Computing
Data Preparation
Physics
Help
Glossary

ATLAS Computing

Problems
Help
Running Jobs (Achelois)
Savannah
Using TWiki

Communication
HyperNews
Simba
Indico (Today)

WorkBooks
Computing
Physics Analysis
Software Development

Tools
AMI
Code Browsers
Tag Collector
AOD & ESD Contents

Search

Getting Started

Analysis WorkBook
Computing WorkBook
For Newcomers
Help
Tutorials

ATLAS e-mail

ATLAS e-mail management top page
ATLAS Hypernews List in e-group
Access to e-groups mail lists

Other Communication

Collaborative Tools
Meetings (Today, All)
Savannah (about)

Documentation

Computing TDR (pdf)
Documentation Management
Glossary

Users and Developers

CernVM - Virtualization of ATLAS Software
AOD & ESD Contents
Athena
Core Software
Debugging Code
Event Data Model
Event Store
Info for Developers
Installing Software (advanced)
Release Recipes
Software Development Workbook
Trouble Running Jobs (Achelois)
Writing Code

Activities

Combined Test Beam
FDR: Full Dress Rehearsal
Physics Validation
Software Validation
S/w Infrastructure Team (SIT)
Upgrade Simulation

Tools

Doxygen (about)
Librarian Tools
Pacman (about)

Code Management

AFS Directories
Code Distribution
Code Management
Nightly Builds (and ATN Testing)
Releases (and Project Builds)
Runtime Testing
Tag Collector (about)

View Code

(help)
BNL Browser
Doxygen Classes (Search)
LXR
TagCollector
View SVN
ViewVC (Deprecated)

Distributed Computing and Grid

AMI: Metadata Interface (about)
ADC: Distributed Computing
Databases
DDM: Distributed Data Management
Ganga
PanDA
AGIS
Pcache
Regional/Local Computing & Tier2
Tier0 Homepage
Tier1 Dataflow
Web Services

Operations

CAF: CERN Analysis Facility
Computing Operations
Computing Operations
Database Deployment
Database Operations
Data Preparation
DDM Operations
Tier-0 Operations

ATLAS Analysis Workbook

Physics Analysis
Combined Test Beam
Physics Validation
Physics Analysis Tools
Physics Analysis Workbook
Statistics Tools in ATLAS

Physics Groups
B Physics WG
Top WG
Standard Model WG
Higgs WG
SUSY WG
Exotics WG
Heavy Ions WG
Monte Carlo WG

Combined Performance
e/gamma
Flavor Tagging
Jet/EtMiss
Tau
Muon

Other Groups
Trigger Alg/Perf/Menu
InDet Tracking Perf

Search

Preface Complete:

- Acknowledgements
- Using the Workbook
- Formatting Rules

Introduction Complete:

- Getting Started
- Overview talks and articles on the ATLAS detector and Physics

Data Formats Complete:

- Introduction
- ATLAS Event Data Model
- Contents of the RDO (Raw Data Object)
- Contents of the ESD (Event Summary Data) and AOD (Analysis Object Data)
- Contents of the Derived Physics Datasets (DPD)
- Contents of Root-tuples
- Contents of the TAGs

Examples

Analysis Examples Complete:

- Introduction
- Important Information (on EDM changes in this release)
- Simple AOD analysis
- **NEW** AOD analysis with DataQuality flags, Luminosity and Trigger Information
- Accessing MC truth information
- Simple AOD Analysis - More exercises
- How to write an algorithm
- How to write an AlgTool
- Services in Athena
- Navigation (and BackNavigation)

GoodRunList twiki

Link to XML files

Good Run Lists for Data

- Check out the Good Run List generator, here !
- Good run lists for data, see here.

News

- 20091207: Update of tutorial will collisions grls.
- 20091126: Addition of section with good run lists for data.
- 20091127: Addition of mini tutorial for using good run lists with first data.

Recommended release and tags

The recommended release for the GoodRunsList package is `15.6.4` or greater. Please use the tags:

```
cmt co -r GoodRunsListsUser-00-00-11 DataQuality/GoodRunsListsUser
cmt co -r GoodRunsLists-00-00-72 DataQuality/GoodRunsLists
cmt co -r LumiBlockComps-00-00-88 LumiBlock/LumiBlockComps
cmt co -r CoolRunQuery-00-01-93 Database/CoolRunQuery
```

(This set of tags has been requested to go into release 15.7.0.)

To install, do:

```
cd $TestArea/Database/CoolRunQuery/cmt/
cmt make; source setup.sh
```

Athena packages and tags you need in your installed area

Tutorial ↓

GoodRunList XML files

Good Run List Generator

This generator allows you to create good run list XML files from configurations that are currently kept on AFS, in `~atlasdqm/grl`.

This can routinely take 1-2 minutes. **Do not keep hitting Submit unless you get a timeout error.**

Using tags:

CoolRunQuery-00-01-93

GoodRunsLists-00-00-69

Configuration file: Tau.tau_2009 (Click to download)

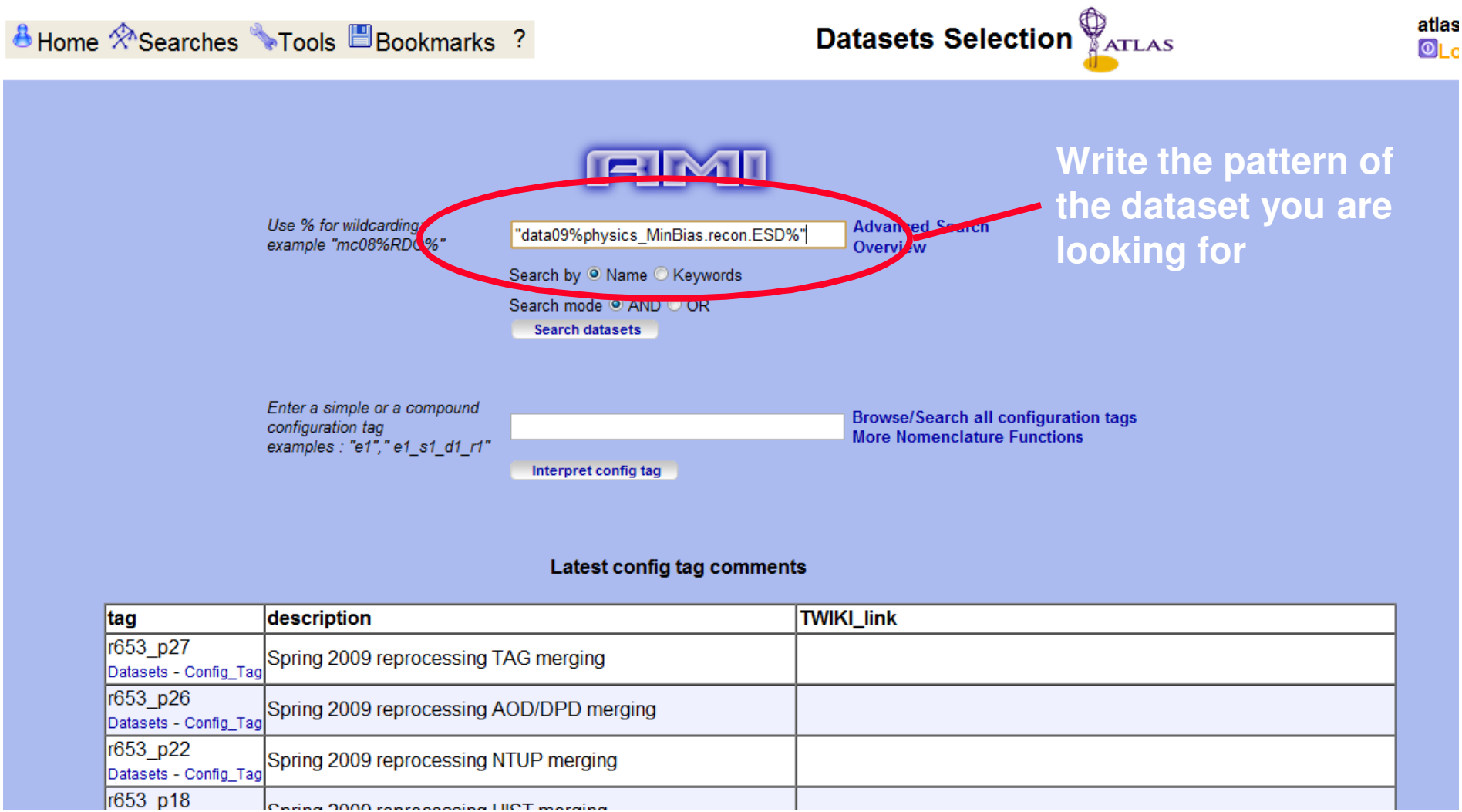
DQ folder: LBSU If in doubt, leave as LBSUMM

COOL tag:

Submit

- Tau.tau_2009
- StandardGRL.SampleConfig
- StandardGRL.minbias_2TeV
- StandardGRL.minbias_900GeV
- StandardGRL.minbias_solon_2TeV
- StandardGRL.minbias_solon_900GeV
- StandardGRL.minbias_solon_windet_900GeV
- StandardGRL.minbias_stable_900GeV
- StandardGRL.minbias_windet_900GeV
- StandardGRL.jetetmiss_jetmetok_windet_900GeV
- StandardGRL.jetetmiss_jetmetok_woindet_2TeV
- StandardGRL.jetetmiss_jetmetok_woindet_900GeV
- StandardGRL.jetetmiss_windet_900GeV
- StandardGRL.jetetmiss_woindet_2TeV
- StandardGRL.jetetmiss_woindet_900GeV

AMI database



Write the pattern of the dataset you are looking for

Tier3g Users' Guide twiki

Argonne NATIONAL LABORATORY

UsAtlasTier3

Log In or Register

UsAtlasTier3 Web

Tier-3g Setup Guide

Introduction

What is a Tier 3g?
The Components of A Tier 3g
How to begin setting up a T3g

T3g Hardware

Deciding what to buy?
Hardware Recommendations?

Preliminaries

TWiki > UsAtlasTier3 Web > Tier3gUsersGuide (31 Mar 2010, TWikiAdminUser)

UNDER CONSTRUCTION 16 March 2010

Tier3g Users' Guide

- Introduction
- Setting Up Your Account
  - Basics
  - Your ATLAS environment
  - Getting ready to run Athena interactively
    - Running on CVMFS athena versions.
    - Accessing SVN code repository at CERN
- Running Athena
  - (Almost) Athena-Version independent HelloWorld example
- Getting sample data and MC files with DQ2
- Submitting to the Grid using pathena
  - Your Grid Certificates
  - Setting up for Pathena
  - Using Pathena to submit to the Grid
- Local Batch Cluster
  - Using pathena to submit to your local batch cluster
  - Local parallel processing on your batch cluster: ArCond
  - Condor
    - Looking at Condor queues

Let's start working…

- Log on ascint0y.hep.anl.gov
- Setup Athena release 15.6.6
  - cd ~belen/Jamboree/2010-03
  - Source SetupATLAS.sh

```
export ATLAS_LOCAL_ROOT_BASE="/export/share/atlas/ATLASLocalRootBase"
alias setupATLAS='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh'

setupATLAS

# set up test area
export ATLAS_TEST_AREA="/users/belen/JamboreeANL/2010-03/15.6.6"

# set up correct version of C++ compiler  (at ANL ASC it is 64-bit slc5)
localSetupGcc --gccVersion=gcc432_x86_64_slc5

# select athena version
source /export/home/atlasadmin/temp/setupScripts/setupAtlasProduction_15.6.6.sh

# access to conditions files and database
export FRONTIER_SERVER="(proxyurl=http://ascvmsquid.hep.anl.gov:3128)(serverurl=http://squid-frontier.usatlas.bnl.gov:23128/frontieratbnl)"
export ATLAS_POOLCOND_PATH="/opt/atlas/conditions/poolcond/catalogue“
```

- Download and compiles the packages that you need for your analysis
- Ready to go!

Prepare your Analysis Code I: Using AnalysisSkeleton

- Similar to what was explained for AODs, only difference is that we'll access cells inside the jets You can add this to the DragonFly class (explained by Rik/Esteban)
- Otherwise you can start from scratch: Make a copy of
  `PhysicsAnalysis/AnalysisCommon/UserAnalysis/src/AnalysisSkeleton.cxx`
  `PhysicsAnalysis/AnalysisCommon/UserAnalysis/UserAnalysis/AnalysisSkeleton.h`
- Create a new class similar to AnalysisSkeleton (I called it MyJetAnalysis)
- Add it to:
  `PhysicsAnalysis/AnalysisCommon/UserAnalysis/src/components/UserAnalysis_entries.cxx`

```
#include "UserAnalysis/AnalysisSkeleton.h"
#include "UserAnalysis/MyJetAnalysis.h"

#include "GaudiKernel/DeclareFactoryEntries.h"
DECLARE_ALGORITHM_FACTORY( AnalysisSkeleton )
DECLARE_ALGORITHM_FACTORY( MyJetAnalysis )

DECLARE_FACTORY_ENTRIES( UserAnalysis ) {
  DECLARE_ALGORITHM( AnalysisSkeleton )
  DECLARE_ALGORITHM( MyJetAnalysis )
}
```

Prepare your Analysis Code II

- Modify your requirements file if you add dependencies to other libraries

```
use JetUtils                    JetUtils-*                    Reconstruction/Jet
```

- We are going to create some histograms with:

  - Number of jets
  - Jet pT
  - Jet eta
  - Jet phi

Only accessible through ESDs

- Number of cells inside a jet
- Jet energy per layer (EM and HAD scale)
- Jet cell energy at EM scale
- Distance between cell and jet axis
- Jet cell energy density

Prepare your Analysis Code III

```
StatusCode MyJetAnalysis::CBNT_initialize() {
                    . . . . .
  //////////////////////
  m_h_jet_njet = new TH1F("jet_njet","jet_njet",20,0,10);
  sc = m_thistSvc->regHist("/AANT/Jet/jet_njet",m_h_jet_njet);

  m_h_jet_pt = new TH1F("jet_pt","jet_pt",100,-10,200);
  sc = m_thistSvc->regHist("/AANT/Jet/jet_pt",m_h_jet_pt);
                    . . . . .
```

Histograms for general properties of the jets

Histograms of jet cell energy for each calorimeter layer

```
for (int ilay=0; ilay<24; ilay++){
    // Cell energy at EM scale
    sprintf(name,"jet_cell_eem[%d]",ilay);
    m_h_jet_cell_eem[ilay] = new TH1F(name,"Layer: "+(*m_LayerName[ilay]),200,-10,100);
    sprintf(name,"/AANT/Jet/jet_cell_eem[%d]",ilay);
    sc = m_thistSvc->regHist(name,m_h_jet_cell_eem[ilay]);

    // Cell deltaR( jet- cell)
    sprintf(name,"jet_cell_deltaR[%d]",ilay);
    m_h_jet_cell_deltaR[ilay] = new TH1F(name,"Layer: "+(*m_LayerName[ilay]),200,0, 2);
    sprintf(name,"/AANT/Jet/jet_cell_deltaR[%d]",ilay);
    sc = m_thistSvc->regHist(name,m_h_jet_cell_deltaR[ilay]);
                    . . . . .
}
```

Prepare your Analysis Code IV

```
/// Method to fill Jet histograms
StatusCode MyJetAnalysis::fillJetHistograms() {
  StatusCode sc;
  MsgStream mLog( messageService(), name() );

  // ----  retrieve jets ---------------
  const JetCollection * jetTES = 0;
  sc = m_storeGate->retrieve( jetTES, m_jetContainerName);
  if( sc.isFailure()  ||  !jetTES ) {
    mLog << MSG::FATAL << "No Jet container found in TDS" << endreq;
    return StatusCode::FAILURE;}
   if (jetTES->size() < 1) return StatusCode::SUCCESS;

 // ---- fill histograms ---------------
  m_h_jet_njet->Fill(jetTES->size());

 // ---------- LOOP over JETS -------------------------
  JetCollection::const_iterator jetItr  = jetTES->begin();
  JetCollection::const_iterator jetItrE = jetTES->end();
  for (; jetItr != jetItrE; ++jetItr) {
    HepLorentzVector p4((*jetItr)->px(),(*jetItr)->py(),(*jetItr)->pz(),(*jetItr)->e());
    HepLorentzVector p4_em ( (*jetItr)->constituent_sum4Mom() );

    //  Fill histograms: Jet pT, eta and phi
    m_h_jet_pt->Fill(p4.perp()/GeV);
    m_h_jet_eta->Fill(p4.eta());
    m_h_jet_phi->Fill(p4.phi());

      . . . . . Continue in next page . . . . .
```

Athena Error Reporting
https://twiki.cern.ch/twiki/bin/view/Atlas/ReportingErrors

Prepare your Analysis Code V

```
// ------- LOOP over cells -----------------
 NavigationToken<CaloCell,double> cellToken;
 (*jetItr)->fillToken(cellToken,double(1.));
 NavigationToken<CaloCell,double>::const_iterator  cbeg = cellToken.begin();
 NavigationToken<CaloCell,double>::const_iterator  cend = cellToken.end();
 mLog << MSG::DEBUG << "Get cell token" <<endreq;
 const double mm3 = millimeter*millimeter*millimeter;
 int ncells = 0;
 double jet_energy_from_cells_emscale = 0.;
 for (; cbeg != cend; ++cbeg)
   {
     ncells++;
     const CaloCell* thisCell = *cbeg;
     const CaloSampling::CaloSample s = CaloSampling::getSampling( *thisCell );
     double cell_weight = cellToken.getParameter(cbeg);
     double cell_eta = thisCell->eta();
     double cell_phi = thisCell->phi();
     double cell_energy  = (thisCell->e())/GeV;
     double cell_wenergy = cell_energy*cell_weight;
     double cell_volume  = thisCell->caloDDE()->volume()/mm3;
     // Fill histograms: Cell energy, DeltaR(cell-jet), Cell energy density
     m_h_jet_cell_eem[s]->Fill( cell_wenergy );
     double deta = p4.eta() - cell_eta;
     double dphi = fabs(JetDistances::deltaPhi(p4.phi(),cell_phi));
     double deltaR = sqrt(std::pow( deta , 2 ) + std::pow( dphi , 2 ));
     m_h_jet_cell_deltaR[s]->Fill( deltaR );
       . . . . . .
}
```

Prepare your job options

- I use a top options that includes RecExCommon:

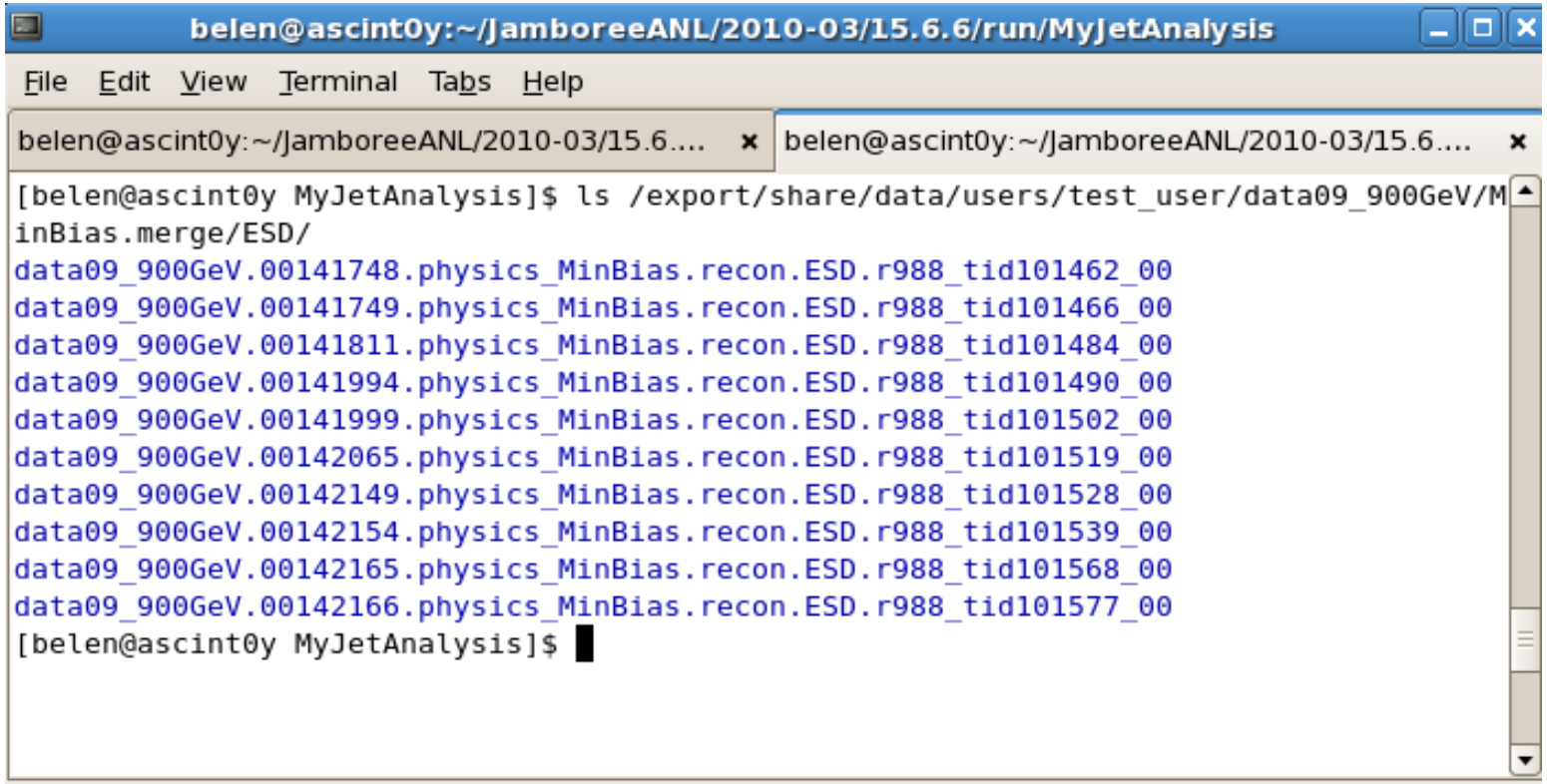
```
EvtMax=-1 ## number of event to process
## include your algorithm job options here ##########################
UserAlgs=[ "GoodRunsListsUser_oneSelection_Belen.py",
           "MyJetAnalysis_jobOptions.py"
           ]
##### SELECT INPUT DATA ##############################################
from glob import glob
INPUTFILES = glob("/data1/chakanau/data/data09_900GeV.*/ESD*pool.root*")
 . . . And some lines more . . .

## Read settings for performance DPD set ESD to true
readRDO = False
readESD = True
readAOD = False
 . . . And some more . . .

## main jobOption - must always be included
include ("RecExCommon/RecExCommon_topOptions.py")
```

Where is the data at ANL?

- XML file that I'm using:

  collisions_stablebeams_minbias_900GeV.xml

- Are in:
  /export/share/data/users/test_users/data09_900GeV/MinBias.merge/ESD



Time to try it . . .