Analysis of ATLAS Cosmic Data: A (not the only) complete approach

Brian T. Martin

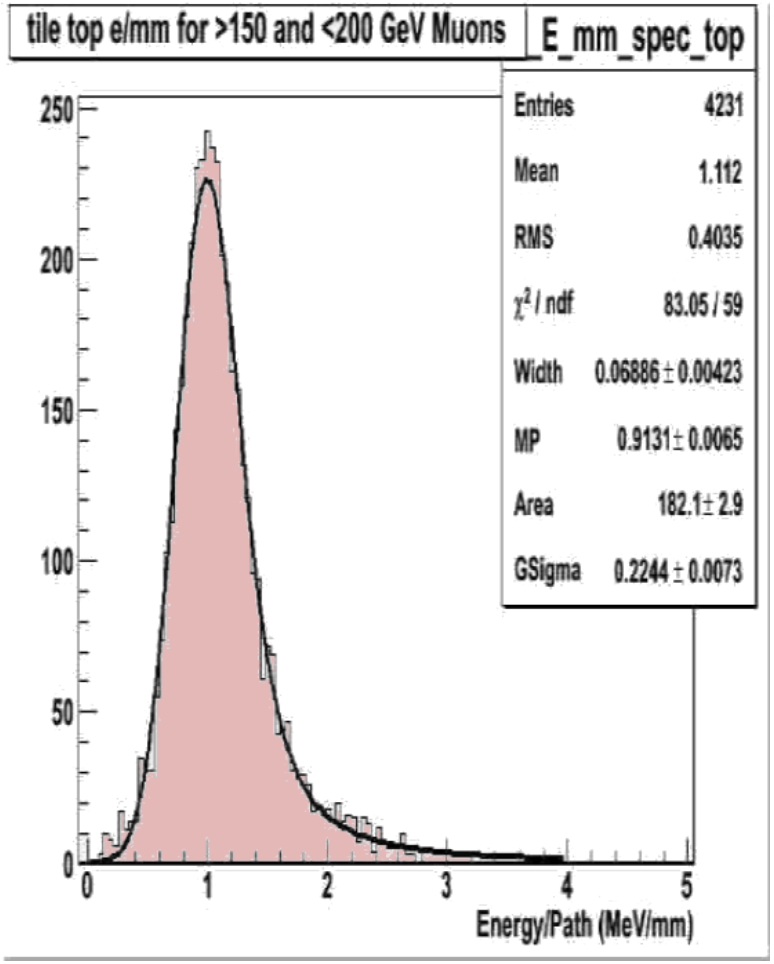
marti347@msu.edu

Michigan State University

September 10, 2009

. . . or how to create this histogram

OK, not really, but this is representative of an interesting histogram one can make by using information from multiple subdetectors in ATLAS cosmic data.



This histogram is stolen from a recent presentation by Rik's summer student, Nathan Gardner. It shows the energy loss per unit path length in the TileCalorimeter for a certain range of cosmic muon p_T as measured by the Muon Spectrometer in 2009 Cosmic data.

ATLAS Cosmic Analysis Overview

While there are many ways to analyze cosmic data, I will focus on the following general prescription.

1. Write a simple ATHENA algorithm to create a custom cosmics ntuple.
2. Test it locally on the ANL cluster. (or any location with 15.4.0 or similar release)
3. Search ATLAS Cosmic data for good quality data. (using Tier0 monitoring histograms and run databases)
4. Submit job to GRID to produce your ROOT-ready cosmic ntuple.
5. Run a ROOT macro analysis on the resulting ntuple to create histograms. (at local institute/laptop)

MuAnalysis: An example of an ntuple-making, ATHENA Algorithm

We will be creating a standard ROOT ntuple similar to what Esteban showed in his talk yesterday.
However, our algorithm will inherit from the Athena Algorithm base class described by Peter on slide 11 of his talk

```
// Original author: Nathan Gardner
class MuAnalysis : public Algorithm  {

  public:
    MuAnalysis(const std::string& name, ISvcLocator* pSvcLocator);
    ~MuAnalysis();
    virtual StatusCode initialize();
    virtual StatusCode finalize();
    virtual StatusCode execute();
    // ...
}
```

To view the entire source code, use ANL ASC's SVN viewer:

http://atlaswww.hep.anl.gov/asc/WebSVN/listing.php?repname=CosmicsAnalysis

MuAnalysis: Information to store in Cosmic Ntuple

We will store the following information in our ntuple:

- Combined Muon quantities
- Muon Spectrometer (MS) quantities
- Inner Detector (ID) track quantities
- Tile Calorimeter muon quantities (TileMuonFitter)
- Other Calorimeter information
- Trigger information

MuAnalysis: Combined Muons

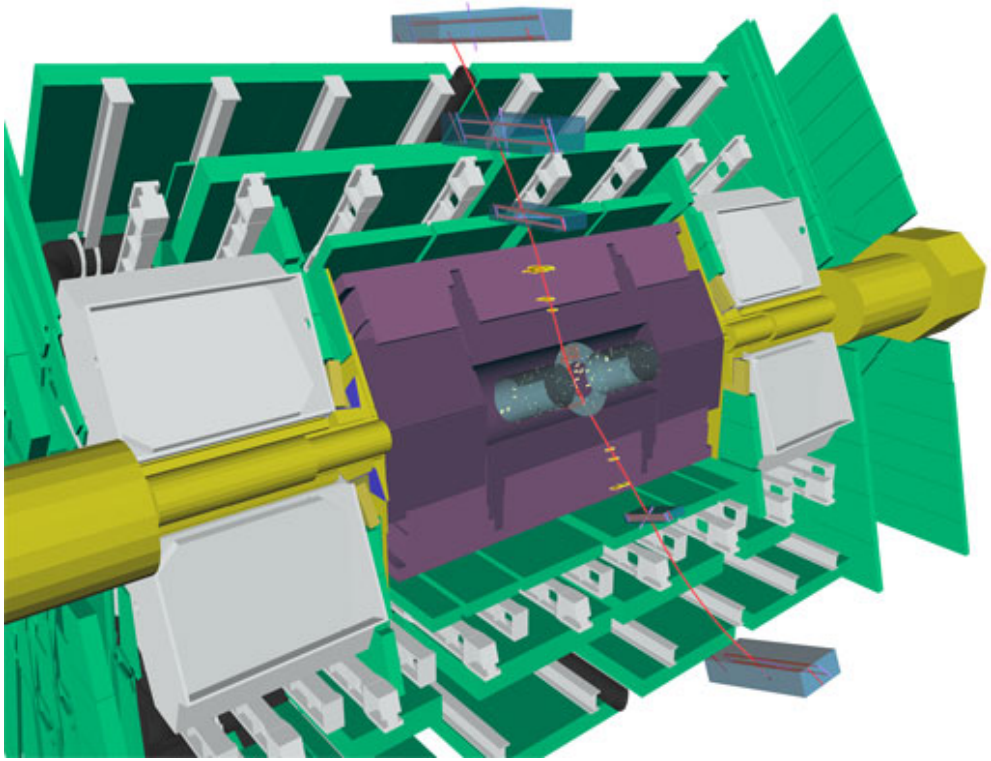


Figure: Combined Cosmic muon showing hits in Muon spectrometer and Inner detector, along with energy deposits in the calorimeter

- Combined Muons taken from 'MuonContainer' in ESD/AOD
- Different algorithms for matching MS tracks to ID tracks and Calo deposits.
- 3 present in ESD/AOD by default:
  1. StacoMuonCollection
  2. MuidMuonCollection
  3. CaloMuonCollection

See:
https://twiki.cern.ch/twiki/bin/view/AtlasProtected/MuonPerformance

MuAnalysis: Deconstructing Combined Muons

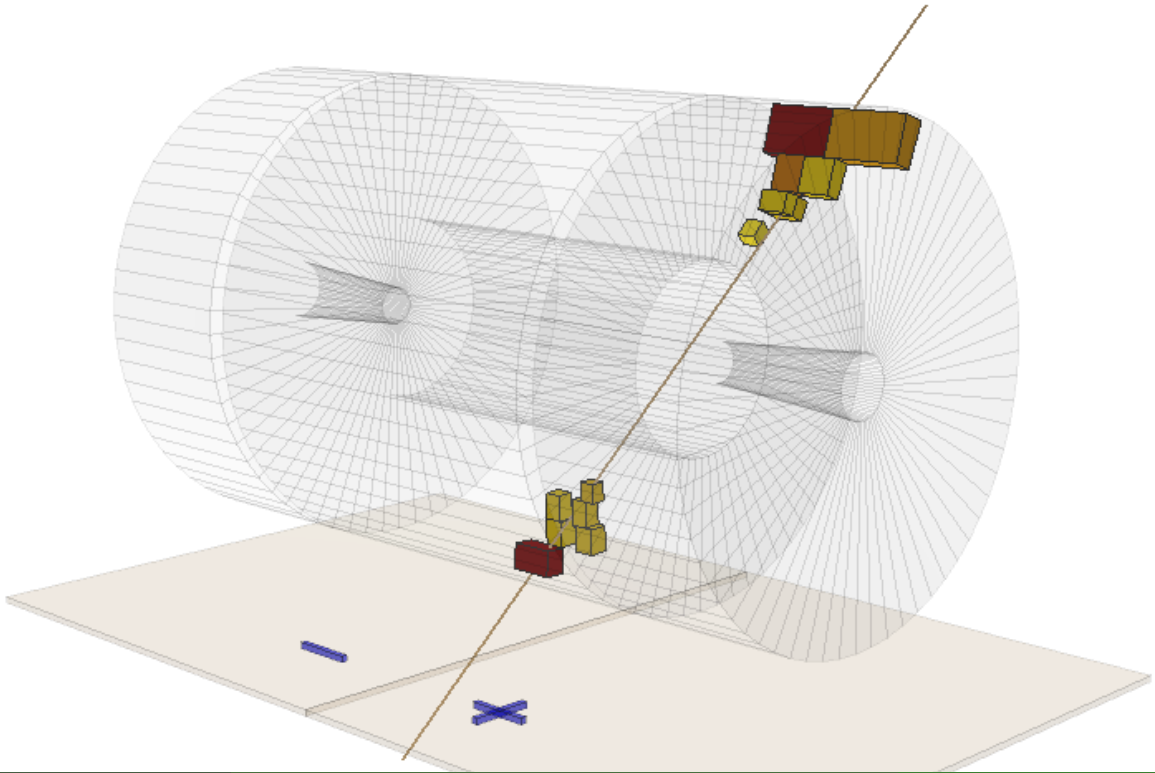
We can ask a Muon (here `itrMuon`) for three different track particles associated with it:

1. Muon Spectrometer Particle (fit of MDT and CSC chambers)
   `itrMuon->muonSpectrometerTrackParticle()`
2. Inner Detector Track Particle (fit of ID components)
   `itrMuon->inDetTrackParticle()`
3. Combined Muon Track Particle (the global fit)
   `itrMuon->combinedMuonTrackParticle()`

All three of these particles are of type: Rec: : TrackParticle, and thus have the same methods available for use (such as: Momentum, eta, phi, and the position of the track at 'Perigee'.)
Note, this 'perigee' is not consistently defined and sometimes needs to be checked with a simple scatter plot.

MuAnalysis: TileMuonFitter Muons

- Another container in ESD/AOD: TileCosmicMuonContainer.
- Tracks from fitting the calorimeter cells with a Hough Transform.
- It 'knows' the cells that created it, thus energy deposited by the muon is known well.
- Has special methods that allow easy calculation of E/mm
- Still provides its direction and its position, given at the $y = 0$ plane.



MuAnalysis: Other information

- This algorithm also loads calorimeter cell and cluster information.
- One pontential extension to the example algorithm, is the matching of muon tracks to calorimeter cells using the ATLAS Extrapolation package:
  https://twiki.cern.ch/twiki/bin/view/Atlas/TrkExtrapolation
- Additionally, this example loads the Trigger information via the Trigger Decision Tool (although currently it is just accessed, not stored.)
- To access a particular trigger decision, say whether L1_MBTSA10 fired or not, one would do within Execute:
  `m_trigDec->isPassed(L1_MBTSA10)`
- For more capabilities of the TriggerDecisionTool, see:
  https://twiki.cern.ch/twiki/bin/view/Atlas/TrigDecisionTool15

MuAnalysis: Compiling and Running

Compiling MuAnalysis

```
cd testarea/15.4.0
svn co file:///users/svn/CosmicsAnalysis/CosmicsAnalysis/trunk CosmicsAnalysis
export AVERS=15.4.0
export TEST_AREA=$HOME/testarea
source /share/grid/app/asc_app/asc_rel/1.0/setup-script/set_atlas.sh
cd CosmicAnalysis/cmt
cmt config
source setup.sh
gmake
```

Running MuAnalysis

```
cd ~/testarea/15.4.0/CosmicAnalysis/run
get_files -jo MuAnalysis_topOptions.py
athena MuAnalysis_topOptions.py
```

MuAnalysis: Configuring via jobOptions

To configure the specific algorithm one must alter this part of the topOptions. For example one can switch the MuonContainer used as input.

MuAnalysis_topOptions.py

```
# Load MuAnalysis
from CosmicsAnalysis.CosmicsAnalysisConf import MuAnalysis
m = MuAnalysis()
m.OutputLevel = WARNING
m.tilemuon_filler = "TileCosmicMuonHT"
#m.specmuon_filler = "StacoMuonCollection"
m.specmuon_filler = "MuidMuonCollection"
m.calocell_filler = "AllCalo"
m.calomuon_filler = "CaloMuonCollection"
m.OutputLevel = INFO
m.OutputFile = "Cosmic_09Data_IDCosmic.root"
topSequence += m
```

MuAnalysis: DB Problem running on Tier 3s

- At tier3s, we use shipped versions of ATHENA that come with a prepackaged set of DB versions.
- Unfortunately the versions of the DB are more suited to simulation.
- This causes problems when running on real data where up-to-date conditions are needed and new tags are requested at runtime.

To solve this I force the Job to use an old tag that can resolve all the folders.
Then when I submit to the GRID, I change the DB version to the desired one.

```
# Conditions
globalflags.ConditionsTag.set_Value_and_Lock('CMCCOND-CSC-00-00-00')  # for local
#globalflags.ConditionsTag.set_Value_and_Lock('COMCOND-REPC-002-13')  # for curre
#include("RecJobTransforms/UseOracle.py") # for current remote DB  !!!!SLOW!!!!
from IOVDbSvc.CondDB import conddb
conddb.setGlobalTag(globalflags.ConditionsTag()
```

MuAnalysis: Configuring via jobOptions

To configure the specific algorithm one must alter this part of the topOptions. For example one can switch the MuonContainer used as input.

MuAnalysis_topOptions.py

```
# Load MuAnalysis
from CosmicsAnalysis.CosmicsAnalysisConf import MuAnalysis
m = MuAnalysis()
m.OutputLevel = WARNING
m.tilemuon_filler = "TileCosmicMuonHT"
#m.specmuon_filler = "StacoMuonCollection"
m.specmuon_filler = "MuidMuonCollection"
m.calocell_filler = "AllCalo"
m.calomuon_filler = "CaloMuonCollection"
m.OutputLevel = INFO
m.OutputFile = "Cosmic_09Data_IDCosmic.root"
topSequence += m
```

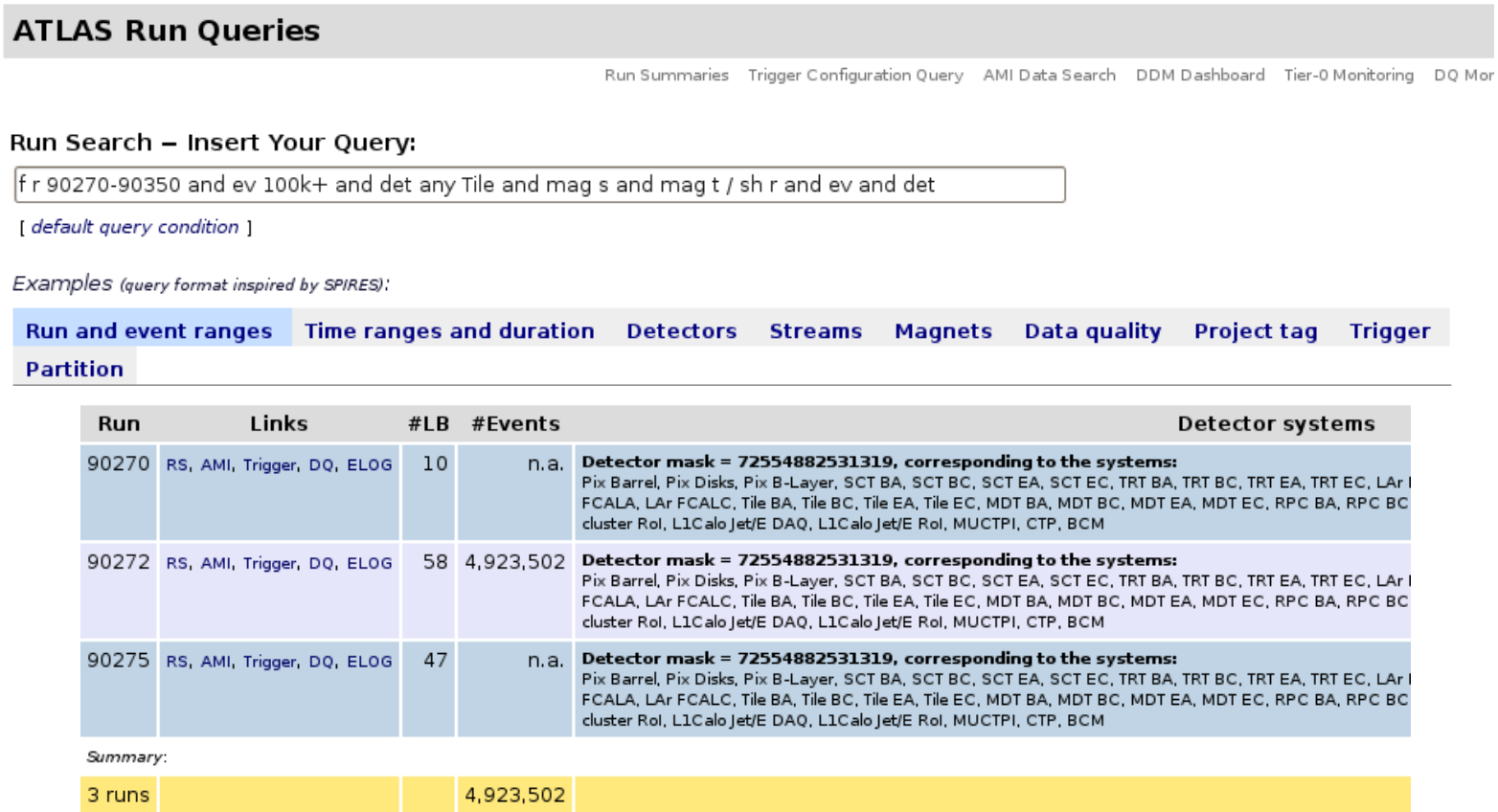
MuAnalysis: Local data for Development

- Develop while running on small datasets here at ANL.
- Submit to GRID for production quality ntuples of fulldataset.
- There are several small datasets available in /data/nas2/users/brianmartin/

```
# POOL input
import AthenaPoolCnvSvc.ReadAthenaPool
svcMgr.PoolSvc.AttemptCatalogPatch=True
import glob
fileList = glob.glob('/data/nas2/users/brianmartin/data08_cosmag.00091900.physics
svcMgr.EventSelector.InputCollections = fileList
```

Finding good runs to analyze

- Take shifts. It gives you a good idea of the overall run conditions.
- Use the run query tool: http://atlas-runquery.cern.ch/
  Example: a range of runs with $> 100,000$ events, any Tile partition, and both magnets on



ATLAS Run Queries

Run Summaries | Trigger Configuration Query | AMI Data Search | DDM Dashboard | Tier-0 Monitoring | DQ Mor

Run Search – Insert Your Query:

f r 90270-90350 and ev 100k+ and det any Tile and mag s and mag t / sh r and ev and det

[*default query condition*]

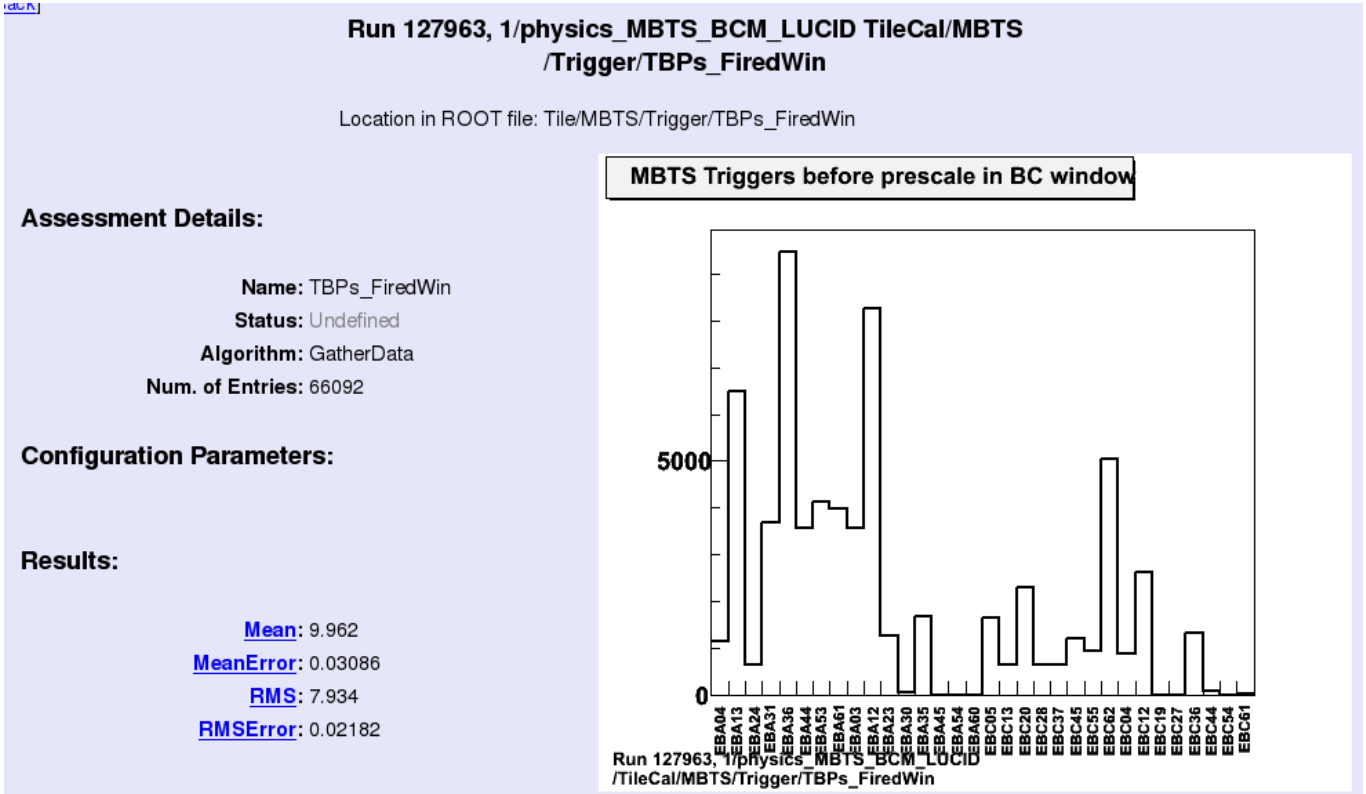
Examples (query format inspired by SPIRES):

Run and event ranges | **Time ranges and duration** | **Detectors** | **Streams** | **Magnets** | **Data quality** | **Project tag** | **Trigger** | **Partition**

Run	Links	#LB	#Events	Detector systems
90270	RS, AMI, Trigger, DQ, ELOG	10	n.a.	**Detector mask = 72554882531319, corresponding to the systems:** Pix Barrel, Pix Disks, Pix B-Layer, SCT BA, SCT BC, SCT EA, SCT EC, TRT BA, TRT BC, TRT EA, TRT EC, LAr I FCALA, LAr FCALC, Tile BA, Tile BC, Tile EA, Tile EC, MDT BA, MDT BC, MDT EA, MDT EC, RPC BA, RPC BC cluster RoI, L1Calo Jet/E DAQ, L1Calo Jet/E RoI, MUCTPI, CTP, BCM
90272	RS, AMI, Trigger, DQ, ELOG	58	4,923,502	**Detector mask = 72554882531319, corresponding to the systems:** Pix Barrel, Pix Disks, Pix B-Layer, SCT BA, SCT BC, SCT EA, SCT EC, TRT BA, TRT BC, TRT EA, TRT EC, LAr I FCALA, LAr FCALC, Tile BA, Tile BC, Tile EA, Tile EC, MDT BA, MDT BC, MDT EA, MDT EC, RPC BA, RPC BC cluster RoI, L1Calo Jet/E DAQ, L1Calo Jet/E RoI, MUCTPI, CTP, BCM
90275	RS, AMI, Trigger, DQ, ELOG	47	n.a.	**Detector mask = 72554882531319, corresponding to the systems:** Pix Barrel, Pix Disks, Pix B-Layer, SCT BA, SCT BC, SCT EA, SCT EC, TRT BA, TRT BC, TRT EA, TRT EC, LAr I FCALA, LAr FCALC, Tile BA, Tile BC, Tile EA, Tile EC, MDT BA, MDT BC, MDT EA, MDT EC, RPC BA, RPC BC cluster RoI, L1Calo Jet/E DAQ, L1Calo Jet/E RoI, MUCTPI, CTP, BCM
Summary:				
3 runs			4,923,502	

Finding good runs to analyze (cont)

- Check the Tier-0 monitoring histograms
- http://atlasdqm.cern.ch:8080/webdisplay/tier0/
  Example: if I was doing a trigger study on the MBTS I might want to check that they were all firing in a certain run.



Submitting to the GRID (pathena)

- See Rik's talk
- One important difference with this example: no use of THISTsvc, so OutputFile must be stated explicitly.
- This can be done with: `--extOutFile my_output.root` (where my_output.root is the ntuple you expect to create.
- Addtionally it is a good idea to plan how many events you want per subjob, and split your job accordingly with: `--split and --nFilesPerJob`
- For more commands see: https://twiki.cern.ch/twiki/bin/view/Atlas/PandaAthena

Now on to the physics

- Assuming the ntuple is produced sucessfully, one can now develop there analysis as a simple ROOT script.
- An example of the analysis that Nathan undertook can be found in the Analysis folder in the CosmicsAnalysis package.
- Particular needs still exist for analyses that span multiple subdetectors and multiple running periods.
- As an example, Nathan was able to observe a Tilecal miscalibration between 2008 data and 2009 data by using Tile Calorimeter, Muon Spectrometer, and Inner Detector data.