A Root Macro to Search for Quiet Tubes in the Sector Thirteen Muon Cosmics Test Data

Craig Dowell
University of Washington, Seattle

Version 0.1.1, 18 October 2009

Abstract

The Sector 13 Muon Cosmics Test has generated a good deal of data. There are lots of histograms saved in various places; but there is no simple, automated way to examine this data with the goal of identifying anomalous tube characteristics. A root macro, "deadtubes.C" has been developed to address this issue by analyzing ntuple files produced by Athena runs. This note describes the operation of version 0.1.1 of this macro and discusses some initial results; particularly that the following quiet tubes are found by running deadtubes.C against the first 10,000 events of combined run 1015:

BML3A13, multilayer 2, layer 3, wire 22;
BOL3A13, multilayer 1, layer 1, tube 31;
BOL3A13, multilayer 1, layer 1, tube 41.

Mezzanine 2 of chamber BOL3A13 is also found to be inactive.

1. Introduction

The goal of "deadtubes.C" is to identify suspiciously quiet drift tubes by examining ntuple output files from Athena runs against the data collected at the Sector Thriteen Muon Cosmics Test.

Three BOL and three BML chambers in ATLAS sector thirteen have been connected since December, 2005. These chambers have been assigned "station" numbers and an ad-hoc Z coordinate identifier called "eta." Table 1 summarizes this naming convention.

Chamber Name	Station	Eta
BML1A13	2	1
BML2A13	2	2
BML3A13	2	3
BOL1A13	4	1
BOL2A13	4	2
BOL3A13	4	3

Table 1: Muon Cosmics Test Chamber Naming

Various runs have been performed, collecting MDT, RPC and combined data. Prior to June, 2006, the triggers were provided by scintillators. After June, the RPCs of sector thirteen have been used for some runs.

A number of these runs have their associated data files published for use in offline analysis. A list of these files can be found at,

https://twiki.cern.ch/twiki/bin/view/Atlas/ListofRunsforOfflineAnalysis

These data files are suitable for input to the Muon Commissioning Software (Athena-based), which produces ntuple files. To see how this process is done, see "Hands on Tutorial with Commisioning Data" (R. Nikolaidou) at,

http://indico.cern.ch/conferenceDisplay.py?confId=4528

There are a significant number of ntuple files that have been generated from the existing commissioning data. These files may be found in

/afs/cern.ch/atlas/ testbeam/muontbh8/scratch07/pit_cosmics/sector13

The deadtubes.C macro scans these ntuple files and searches for tubes with suspiciously low hit rates, assuming that such tubes are dead or "almost dead." During the process, deadtubes.C collects various statistics which are also displayed since they may aid in an analysis of the run.

The output of deadtubes.C is divided into four sections:

1. A "banner" section with version and ntuple file information.

2. A "Summary of inactive mezzanine cards" section. Groups of 24 tubes corresponding to inactive mezzanine cards are reported once in the first section of the output.

3. A "Summary of hedgehog statistics" is output that is composed of the average hit rates for the tubes in each active mezzanine / hedgehog card.

4. A "Summary of quiet tubes" is output that lists all of the tubes that have hit counts below a given threshold level (currently defaults to five).

2. Running the Macro

The current release of the macro can be found on AFS in the directory,

/afs/cern.ch/user/c/cdowell/public/root/release

There is an example ntuple file in the directory called 1015.cd.10000evt.ntuple.root, that is an ntuple file with 10,000 events extracted from the combined RPC and MDT run 1015.

There are, of course, several ways to run the macro. None requires graphical output and so Root may be run with the –b (batch mode) option. The first option is just to run the macro with the default file name (which is the name shown above). This is done by executing the following command.

```
> root -b -q deadtubes.C
```

You type the text in **bold** (or **bold blue** if you have a color version of this note).

You will see the output of the macro (which is discussed below) scroll by on the screen and then you will return to the Linux prompt. You can save the output to a text file if you want by redirecting standard output in the usual way.

```
> root -b -q deadtubes.C > 1015.out
```

If you continue with this approach, you will have to edit the deadtubes.C macro to point the macro to another file every time you get a new ntuple file. This is not recommended. The preferred approach is to run the macro interactively.

```
> root -b
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.12/00      10 July 2006   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

Compiled on 11 July 2006 for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.13, June 8, 2006
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0] .L deadtubes.C
root [1] deadtubes(“1015.cd.10000evt.ntuple.root”);
```

At this point, the output of the program begins scrolling on the screen and you are returned to the Root prompt when the macro terminates. If you want to save the output to a text file, you can do so in the following way:

```
root [0] .L deadtubes.C
root [1] deadtubes("1015.cd.10000evt.ntuple.root"); > 1015.out
root [2]
```

The output will be redirected to the file “1015.out”.

3. The Output

As mentioned above, the output is arranged in four sections. The following descriptions are based on the output file “1015.out” that was generated by the last method of the previous secion.

3.1 The Banner Section

The first section of the output is shown below.

```
***** Version:  deadtubes.C 0.1.1 Sun Aug 13 14:34:18 CEST 2006 *****
***** Input: 1015.cd.10000evt.ntuple.root *****
***** Processing 10000 events *****
```

Figure 1: Example of a Banner Section

The banner begins with the version of the deadtubes.C macro that generated the output (version 0.1.1 that was last changed on August 13, 2006). The name of the input file is also displayed. This is simply the string that you entered as the filename when you ran the program. Finally, the banner displays the number of events that were examined. This defaults to the number of events in the file.

3.2 The Inactive Mezzanine Cards Section

An example of the second section of the output is shown in Figure 2.

```
====================================
Summary of inactive mezzanine cards
====================================
s e m  w mezz flag
- - - -- ---- ----
2 1 1 49   12
2 1 2 49   13
2 1 2 49   13 ????
4 1 1 33    8
```

Figure 2: Example of an Inactive Mezznine Cards Section

The section title appears, followed by a key to the displayed values. This key is followed by the list of mezzanine cards that appear to be inactive. In order to determine if the mezzanine card is inactive, we look at all of the tubes in the first layer associated with the mezzanine. If they all have zero counts, we assume that the mezzanine is inactive and the other layers will also have zero counts. The program does look at the wires corresponding to the other layers in the mezzanine / hedgehog combination, and if it finds a non-zero hit count, it prints a new line with the flag set to “????” that indicates this unexpected non-zero hit count condition.

Taking the first line as an example,

```
2 1 1 49   12
```

The first four numbers are (as described by the key) the address of the mezzanine card, consisting of the station, eta, multilayer, and wire numbers. The wire number is the first wire associated with the mezzanine card. For example, wires 1 – 8 might be associated with a particular mezzanine card, while wires 9 – 16 would be associated with another. In this example, wire 49 is associated with mezzanine card number 12.

The fifth number, as you may have discerned, is the mezzanine number corresponding to the address composed from the first four numbers. See

http://doc.cern.ch//archive/electronic/cern/others/atlnot/INT/muon/muon-int-2006-005.pdf

and

http://atlas.web.cern.ch/Atlas/GROUPS/MUON/TDR/pdf_final/naming.pdf

for details on naming conventions for chambers.

3.3 The Hedgehog Statistics Section

An example of the next section of the output is shown below. The section title appears, followed by a key to the displayed values. This key is followed by the list of the average number of hits on all of the tubes of the specified hedgehog (inactive mezzanine cards are not listed).

```
==============================
Summary of hedgehog statistics
==============================
s e m  w mezz average
- - - -- ---- -------
2 1 1  1    0  58.708
2 1 1  9    2  79.125
2 1 1 17    4  76.542
4 3 1  1    0  90.667
4 3 1 17    4  89.208
4 3 1 25    6  79.542
4 3 1 33    8  99.792
4 3 1 41   10  38.292
```

Figure 3: Example of a Hedgehog Statistics Section

Taking the first line as an example

```
2 1 1  1    0  58.708
```

The first four numbers are (as described by the key) the address of the hedgehog / mezzanine, consisting of the station, eta, multilayer, and wire numbers. The mezzanine card number corresponding to this address is displayed, followed by the average number of hits on the 24 tubes composing the given hedgehog.

3.4 The Quiet Tubes Section

This section of the output is shown below. As in the previous section, the section title appears, followed by a key to the displayed values. Note that the threshold for detection of a quiet tube is printed in the title. In this case, the program flags all wires that have a hit count less than or equal to a hard-coded value defined by the program constant kTHRESHOLD_OF_DEATH, which is set to 5 in this case.

```
======================================
Summary of quiet tubes, threshold = 5
======================================
s e m l  w hits mezz
- - - - -- ---- ----
2 3 2 3 22    0    5
4 3 1 1 31    0    6
4 3 1 1 41    3   10
```

Figure 4: Example of a Quiet Tubes Summary Section

Taking the first line as an example,

```
2 3 2 3 22    0    5
```

The line begins with a tube address consisting of the station, eta, multilayer, layer and wire. The next number is the number of hits on the tube corresponding to the wire number. The mezzanine number to which the suspect tube belongs is displayed as a convenience.

If you look at the last line in the output, you will notice that quiet tubes do not necessarily have zero counts. The tube at address "4 3 1 1 41" has three hits. This is significantly different than the number of hits recorded on its neighbors. To see this, look at the hedgehog statistics and find the corresponding hedgehog.

The easiest way to do this is by scanning down the hedgehog statistics list (Figure 3) for the hedgehog addressed by "4 3 1." Ignore the tube number and look for mezz 10, and you will find that the average for this hedgehog neighborhood is 38.292 hits.

```
4 3 1 41   10  38.292
```

It is statistically very unlikely that we should find three hits on a tube with an expected average of 38. There is something clearly amiss with this tube, but it does not register zero hits. Finding "dead" tubes is a bit more subtle than we originally expected.

4. Initial Results

In order to generate ntuple files with sufficient statistics for our purposes, we ran the Muon Commissioning Athena Software ourselves on the combined MDT + RPC run 1015 data stored on Castor, processing only the MDT data, and extracted the first 10,000 events The average number of hits per tube grouped by hedgehog is shown in Figure 5.

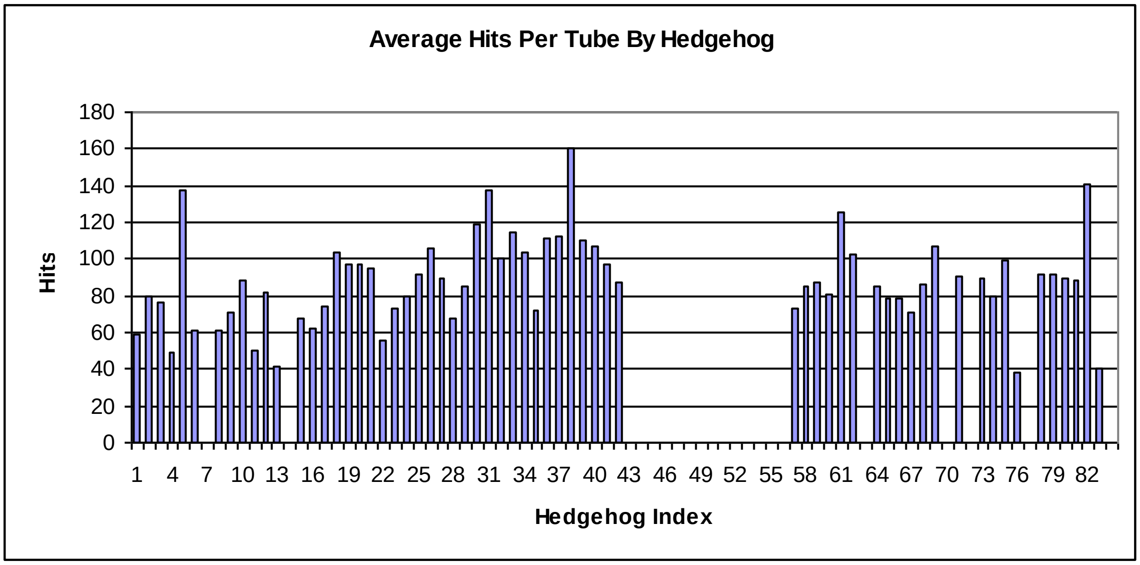

Figure 5: Hedgehog Statistics Histogram for file 1015.cd.10000evt.ntuple.root. Hedgehogs are numbered sequentially starting at station 2, eta 1.

The output of this section of deadtubes.C is shown in Figure 6, below.

```
======================================
Summary of quiet tubes, threshold = 5
======================================
s e m l  w hits mezz
- - - - -- ---- ----
2 3 2 3 22    0    5
4 3 1 1 31    0    6
4 3 1 1 41    3   10
```

Figure 6: Quiet Tubes Identified from file 1015.cd.10000evt.ntuple.root

The macro has identified three tubes as being questionably quiet – two with zero hits, and one with three hits. Note that the tube addressed[1] by “4 3 1 1 31” shows three hits. A nonzero number three is surprising, but may be due to electronic nose crosstalk, etc.

If the average hit rate per tube is about 20, Poisson statistics suggest roughly a fifteen percent probability of finding at least one tube statistically varying down under the bad tube cut of five hits or fewer. It should be noted that the average hit rates in the hedgehog statistics can be distorted by the presence of noisy tubes.

[1] . The station and eta can be mapped to the chamber identifier BOL3A13 using Table 1.