

Table of Contents

Hotspot Finder for Atlas DQM Web Display.....	1
Location.....	1
Using artifact_correlator.py.....	1
Example Usage.....	1
Code Structure.....	1
Importing for Other Uses.....	2
Planned Changes.....	2
Other Pages.....	2

Hotspot Finder for Atlas DQM Web Display

A python script was created for finding hotspots found in the tile and LAr calorimeter Eta, Phi histograms and then finding which hotspots in other histograms were and were not affected by the LAr hotspot. For all purposes of this wiki and the python code itself, hotspots and artifacts are equivalent in usage and meaning.

Location

- The Python script can be found in:
`/afs/cern.ch/user/t/tcartwri/public/atlas_dqm/artifact_correlator.py`
- Example Delta-R_Squared plots can be found in:
`/afs/cern.ch/user/t/tcartwri/public/atlas_dqm/`
- `artifact_correlator.py` requires Python 2.7+ and ROOT to function

Using artifact_correlator.py

1. Download `artifact_correlator.py`
2. Execute file with either `./artifact_correlator.py` or `python artifact_correlator` and arguments. Arguments:
 - ◆ `-h, --help` Displays a help message for the script
 - ◆ `-i, --index` A list of index histogram names passed as a string and separated with whitespace
 - ◆ `-t, --test` A list of test histogram names
 - ◆ `-p, --plot` Plots Delta-R, off by default
 - ◆ `-r, --run_number` The run number as a string or int
 - ◆ `-d, --cut` The Delta-R cutoff, default of 0.16

Example Usage

```
./artifact_correlator.py -i  
"Global/HotSpotFinder/Detectors/Tile/tileCellEneEtaPhiSampAAnyPhysTrig@HotSpotFinder  
Global/HotSpotFinder/Detectors/Tile/tileCellEneEtaPhiSampBAnyPhysTrig@HotSpotFinder" -t  
"Global/HotSpotFinder/Objects/Jets/OccupancyEtaPhi@HotSpotFinder Tau/tauPhiVsEta" -r  
281385 -d 0.17
```

Code Structure

Currently, `artifact_correlator` is divided into 4 main section: Global, artifinder class, artifact class, and executor

- The Global section is where global variables used by the script are generated such as `RUN_NUM`, `SRC`, `STREAM`, `HISTOS`, and `INDEX_HISTOS`
- The `artifinder` class contains all of the methods used to correlate the hotspots found in `INDEX_HISTOS` and those found in `HISTOS`
- The `artifact` class is a convenient object class to store the information generated by `get_dqmF_all_results`
- The executor is where an instance of the `artifinder` class is created using information from the Global section. Methods found in the `artifinder` class can then be called on the object instance.

Importing for Other Uses

1. From `artifact_correlator.py` import `artifinder`, `artifact`
2. Create `artifinder` object in the form of `artifinder(INDEX_HISTOS, TEST_HISTOS, STREAM, SRC, RUN_NUM, D_SQUARE_CUT)` where `INDEX_HISTOS` and `TEST_HISTOS` are 1-D arrays/lists containing the histogram names as strings
3. `artifinder.find_artifacts()` will return a list containing the names of the histograms affected by the index histograms

Planned Changes

- Add default Loose and Tight cut values
- Add cold spot finder algorithm

Other Pages

ClosureCompiler

This topic: [Sandbox](#) > [ThomasCartwrightSandbox](#)

Topic revision: r7 - 2016-01-15 - [ThomasCartwright](#)



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)