cms-l1t-analysis Documentation

Release 0.1.1

kreczko,benkrikler

August 02, 2017

Contents

Contents:

CHAPTER 1

What is cms1lt?

cmsl1t is a python package for Level 1 Trigger analysis for the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider (LHC).

Goals of the package

What is not included.

CHAPTER 2

CMS L1T Analysis Tutorial

Getting started

cmsl1t requirements

Install cmsl1t

Run local examples

Run examples on HTCondor

Run examples on the grid

Configuration

This project uses configuration files in the YAML format to define the workflow.

Disclaimer: We are still in alpha, this section is likely to change.

A config file consists of muliple sections. The `general` section describes the version and name of the config.

```
general:
  version: 0.0.1
  name: Benchmark
```

The input section describes the data that is to be processed and might be changed in the near future. The first subsection, `files`, is a list of files that can be either relative paths, absolute paths or global paths (e.g. xrootd) and can include wildcards.

```
input:
  files:
    - data/L1Ntuple_*.root
```

The second subsection, `sample` is used to describe the data: The name of the dataset, the title and the run number. The name is likely used in file and histogram names, while the title is meant to be used in string representations (e.g titles/legends of histograms). If pileup reweighting is required, the `pileup_file` parameter needs to be set.

```
input:
  ...
  sample:
    name: Data
    title: 2016 Data
    pileup_file: ""
    run_number: 276243
```

The trigger subsection describes which trigger is to be used for this dataset. As the sample name and title, the trigger counterparts play a similar role.

```
input:
  ...
  trigger:
    name: SingleMu
    title: Single Muon
```

The `analysis` section describes which analyzers are to be run. Global parameters include flags and binning for the analyzers (`do_fit`, `pu_bins`). These can also be specified later separately for each analyzer if required.

```
analysis:
  do_fit: False
  pu_bins: 0,13,20,999
```

The analyzers subsection of `analysis` is a list of all analyzers to be run. These analyzers have to satisfy the same API as `cmsl1t.analyzers.BaseAnalyzer` and be visible in the *PYTHONPATH*.

```
analysis:
  ...
  analyzers:
    - cmsl1t.analyzers.demo_analyzer
```

Modifiers are a way to enrich the event content by attaching objects to the event itself. E.g. `cmsl1t.recalc.met.l1MetNot28` reads in `event.caloTowers` and creates a new object, `event.l1MetNot28`, that can then be accessed by all analyzers.

```
analysis:
  ...
  modifiers:
    - cmsl1t.recalc.met.l1MetNot28:
        in: event.caloTowers
```

```
        out: event.l1MetNot28
    - cmsl1t.recalc.met.l1MetNot28HF:
        in: event.caloTowers
        out: event.l1MetNot28HF
```

Next, you can specify if you want progress information (e.g. a progress bar) and how often this information is updated (`report_every` in units of events).

```
analysis:
  ...
  progress_bar:
    report_every: 1000
  # or to switch it off
  # progress_bar:
  #   enable: False
```

And finally the output section describes where the output, usually ROOT files, is stored. The `template entry is composed of a list of paths that are joined to create the full output file. The template expects the following named parameters:

- `date`
- `sample_name`
- `run_number`
- `trigger_name`

which are automatically filled by the config parser

```
output:
  # template is a list here that is joined (os.path.join) in the config
  # parser
  template:
     - benchmark/new
     - "{date}_{sample_name}_run-{run_number}_{trigger_name}"
```

So a complete example would look something like that:

```
version: 0.0.1
name: Benchmark

input:
  files:
    - data/L1Ntuple_*.root
  sample:
    name: Data
    title: 2016 Data
  trigger:
    name: SingleMu
    title: Single Muon
  pileup_file: ""
  run_number: 276243
```

```
analysis:
  do_fit: False
  pu_type: 0PU12,13PU19,20PU
  pu_bins: 0,13,20,999
  analyzers:
    - cmsl1t.analyzers.demo_analyzer
  modifiers:
    - cmsl1t.recalc.met.l1MetNot28:
        in: event.caloTowers
        out: event.l1MetNot28
    - cmsl1t.recalc.met.l1MetNot28HF:
        in: event.caloTowers
        out: event.l1MetNot28HF

output:
  # template is a list here that is joined (os.path.join) in the config parser
  template:
     - benchmark/new
     - "{date}_{sample_name}_run-{run_number}_{trigger_name}"
```

Creating histograms

Adding your own analyzer

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

CHAPTER 4

API Reference

Python Module Index

C

Index

C