On-Demand Distributed Computing Workflow for Physics Analysis at the CMS Experiment

Diyaselis Delgado López[1,2]

[1]CERN, Geneva, Switzerland

[2]University of Puerto Rico, Mayagüez, Puerto Rico

December 12, 2018

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science in the High Energy Physics Unit of the Department of Physics

This project was performed under the supervision of Dr. Kati Lassila-Perini, from Helsinki Institute of Physics, and Dr. Sudhir Malik, from University of Puerto Rico Mayagüez

Abstract

The CMS experiment is a strong contributor to the CERN Open Data Portal, the CMS Open Data project aims to release real data collected from proton-proton collisions at the LHC to the public. In doing so, it publishes research level data together with environment, software and instructions on how to use it. These data accompanied with their proper analysis can be used scientific research outside the CMS collaboration. This project takes part in developing and testing simplified examples of open data while providing a connection to analysis preservation and a new reproducible research data analysis platform. The purpose of this work is to improve and build upon the CMS Open Data release, by allowing users to study higher energy 13 TeV collisions from preserved analyses.

Keywords: Open Data, analysis preservation, reproducibility, computing, workflows

Contents

1 Introduction

This thesis arises from the ongoing work with experimental data from proton-proton collisions analyzed from the CMS (Compact Muon Solenoid) [1] experiment at the Large Hadron Collider (LHC) [2]. Here, experimental data is analyzed to study the fundamental structure of particles and to search for evidence of new particle physics. This particular project is based in High Energy Physics analysis for public use in scientific research, and involves using frameworks produced for analysis preservation and reproducibility.

This paper is organized as follows: in Section II we introduce proton-proton collisions, and we briefly describe the CMS detector in section III. In Section IV, we outline and define our preservation and reproducibility platforms. We then review analyses tested towards open experimental data in Section V and present event analysis results in Section VI. Finally, our conclusions are summarized in Section VII.

1.1 Proton-proton (pp) collisions at LHC

The LHC is the largest particle accelerator in the world, stationed at CERN [3], the European Organization for Nuclear Research, on the French-Swiss border near Geneva. It is located in an underground tunnel 27 km in circumference. The LHC accelerates high-energy particle beams near to the speed of light, before its components collide together at a primary vertex. Each proton-proton collision between constituents is called an event. The particles produced in events are reconstructed in detectors scattered across the LHC, and used to infer the quantum-mechanical process that occurred in the collision. First, the LHC collided protons at a center of mass energy of 7 TeV, upgraded in 2011 to 8 TeV, and it is currently running at 13 TeV.

1.2 The CMS experiment

The CMS experiment is a detector placed at one of the pp collision points in the LHC ring. Collision products travel out from the collision point into the detector. CMS can directly detect muons, electrons, photons and hadronic jets. These jets are clusters of charged particles that result from the formation of hadrons from quarks and gluons produced in collisions. The CMS detector is

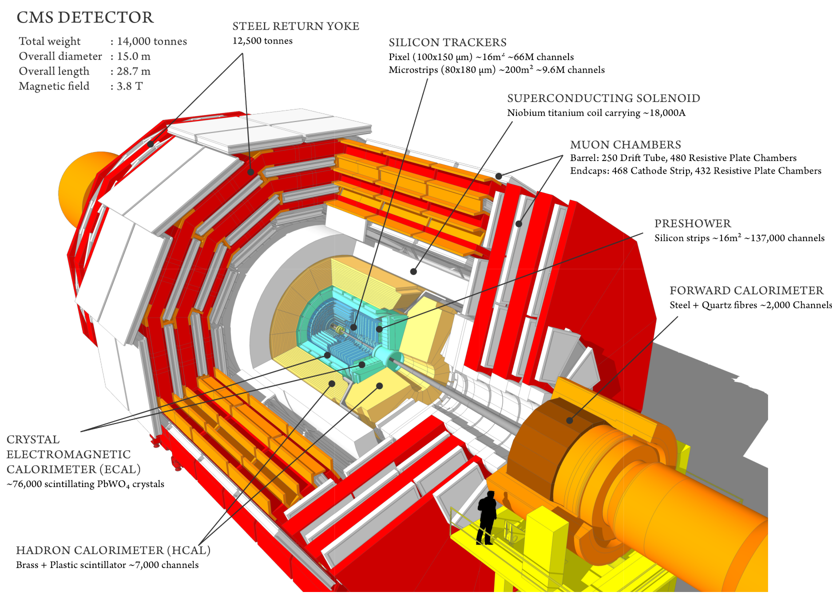

Figure 1: A schematic diagram of a cut-away section of the ATLAS detector. Detector dimensions are shown and people can be seen on site for a sense of scale.

shown in Figure 1. Towards the center of the detector are the trackers, which measure particle positions so that their momenta can be calculated. The CMS detector [4] consists of a superconducting solenoid providing a uniform magnetic field of 3.8 T in the core, equipped with silicon pixel and strip tracking systems ($|\eta| < 2.5$), these magnets curve the tracks of charged particles to enable momentum and charge measurements, ideally to identify the particles resulting from the collisions. Outside the trackers is a lead tungstate crystal electromagnetic calorimeter (ECAL), which measure energy deposits from electrons and photons. Further out is the brass- scintillator hadronic calorimeter (HCAL) covering $|\eta| < 3.0$ to measure jets of hadrons. The steel return yoke outside the solenoid is instrumented with gas ionization detectors used to trigger and identify muons up to $|\eta| < 2.4$.

The reconstruction and the identification of individual particles in the collision event relies on the the Particle Flow algorithm [5, 6] that uses the information from all CMS sub-detectors to provide a list of mutually exclusive categories: charged hadrons, neutral hadrons, photons, muons, and electrons.

2 Analysis preservation and reproducibility platforms

The CERN Open Data portal (CODP) [7] is the gateway to an increasing range of data produced at the LHC experiments. It distributes the preserved product from various research studies, including accompanying software and documentation which is needed to understand and analyze the data being shared. The CMS experiment releases and distributes through CODP complete data and software to allow a full scientific analysis, and provides some simplified data formats for education use. Through the portal, CMS provides primary datasets, full reconstructed collision data with no other selections, simulation data, and analysis tools such as downloadable Virtual Machine (VM) images with the CMS software environment and ready-to-use online applications, like event display or simple histogramming software.

CMS also provides some analysis examples on the portal. These examples are taken here as use cases for analysis preservation. The analysis record on CODP consists of data (e.g. datasets, code, results) and metadata (e.g. analysis name, contact persons, publication). This information can then be structured to represent the analysis workflow steps on the CERN Analysis Preservation platform (CAP) [8]. The CAP service aims to enable physicists to preserve information as it is produced by making it easier to describe, preserve, find, exchange and export information in a fast paced scientific environment. CAP does not require any change in the physicists' individual arrangements or the terminologies used in the different LHC collaborations. Moreover, this framework is developed to address the need for the long-term preservation of the data analysis process, in order to enable future reproducibility of research results.

In CAP, information is preserved with the aim of reusing it, for example in ReANA[9], a reusable and reproducible research data analysis platform. In ReANA, a preserved analysis can be rerun starting from structured input data, analysis code, containerized environments and computational workflows so that the analysis can be established and run on remote computer clouds. ReANA was developed to target the use case of particle physics analyses, but is applicable to any scientific discipline. The system paves the way towards reusing and reinterpreting preserved data analyses even several years after the original publication.

3 Physics Analyses on CMS Open Data

3.1 Higgs-to-four-lepton analysis example

The example studied is a simplified reimplementation of the Higgs discovery using CMS open data inputs, both raw data and Monte-Carlo simulations, from 2011-2012 [10]. The data used for the example is actual, meaningful data from the CMS experiment that confirmed the existence of this elusive particle, which then resulted in a Nobel prize. The example contains multiple levels, from very simple to a complete analysis. In order to structure the analysis for ReANA, it must be described under four main categories: inputs, environment, workflow, and code. Then, taking into consideration the "Level 3" of the example, an intermediate stage where part of the analysis is runned and the rest is taken from preprocessed data files, we can describe the fabricated output for the discovery of Standard Model Higgs. The method used to create the analysis firstly derives from some theoretical background, then proceeds to make measurements and testing those measurements to compare with established assumptions.

According to the Standard Model (SM), one of the ways the Higgs boson can decay is by first creating two Z bosons that then decay further into four leptons (i.e. electrons, muons, etc.), as shown in Figure 2. The Higgs boson is a scalar particle predicted by the Standard Model of electroweak interactions [11, 12, 13], responsible for the mechanism of spontaneous symmetry breaking that allow fermions and gauge bosons to acquire mass. The theory does not predict the mass of Higgs boson but it can be obtained experimentally from meticulous assumptions; for example, four lepton decay is very dominant in some mass regions and as so it guides this analysis.

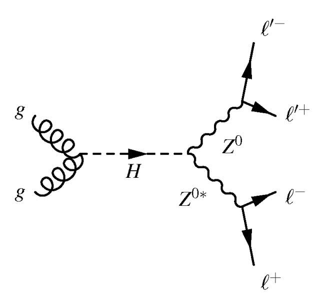

Figure 2: Feymann diagram of a Higgs production process and ZZ decay.

The example is worked on the LXPLUS service [14], lxplus.cern.ch, an interactive logon service to Linux for all CERN users. The cluster LXPLUS consists of public machines provided by the IT Department for interactive work. The example also works with the Open Data virtual machine image described in the CODP for 2011 and 2012 CMS open data[15]. Setting up the computing environment implies building a specific software area, in this case, CMSSW_5_3_32. This assures that there is a working ROOT setup and all CMS code framework is at our disposal. Following step is to add analysis code for both the data example and the Higgs simulation example. Finally, we need to capture the analysis workflows, containing the commands we have to run, to obtain the final plot. The ReANA platform captures the analysis workflows and runs the commands remotely to obtain the outputs of any example.

After structuring the analysis and stating any initial conditions, a physicist only needs to create the computational workflows ReANA follows, these include workflows describing: inputs, command steps, and overall schematics. For the Higgs to 4 lepton example, the inputs were listed as so:

Cert_190456-208686_8TeV_22Jan2013ReReco_Collisions12_JSON.txt
DY1011.root
DY1012.root
DY101Jets12.root
DY50Mag12.root
DY50TuneZ11.root
DY50TuneZ12.root
DYTo2mu12.root
DoubleE11.root
DoubleE12.root
DoubleMu11.root
DoubleMu12.root
HZZ11.root
HZZ12.root
TTBar11.root
TTBar12.root
TTJets11.root
TTJets12.root
ZZ2mu2e11.root
ZZ2mu2e12.root
ZZ4e11.root
ZZ4e12.root
ZZ4mu11.root
ZZ4mu12.root

For all these input datasets, the workflow follows a simple yaml format, "*input.yaml*", in which yaml is a known data serialization standard for all programming languages [16]:

```
inputs:
  class: Directory
  path: ../inputs
code:
  class: Directory
  path: ../code
```

Other elements of the analysis called by the *input.yaml* are the code files used. In this case, to reproduce the Higgs example it was necessary to add the following files: BuildFile.xml, HiggsDemoAnalyzerGit.cc, demoanalyzer_cfg_level3data.py, demoanalyzer_cfg_level3MC.py, M4Lnormdatall_lvl3.cc.

The analysis will consist of two stages. In the first stage, the original collision data (using demoanalyzer_cfg_level3data.py) and simulated data (using demoanalyzer_cfg_level3MC.py) will be processed for one Higgs signal candidate with with reduced statistics[17]. For the second stage, the outputs are created, in this case the results are plotted using M4Lnormdatall_lvl3.cc. Subsequently, the next workflow describes the commands and steps, mentioned above, to execute the analysis. The format of such workflow is CWL[18], Common Workflow Language, a specification for describing analysis workflows and tools in a way that makes them portable and scalable across a variety of software and hardware environments. The first stage is divided into two workflows, one for raw data and another for Monte-Carlo simulations, named "*step1data.cwl*", as shown on the left, and "*step1mc.cwl*", respectively.

```
cwlVersion: v1.0
class: CommandLineTool

baseCommand: /bin/zsh

requirements:
  DockerRequirement:
    dockerPull:
      clelange/cmssw:5_3_32
  InitialWorkDirRequirement:
    listing:
      - $(inputs.code)
      - $(inputs.inputs)

inputs:
  inputs:
    type: Directory
  code:
    type: Directory

stdout: step1data.log

outputs:
  step1data.log:
    type: stdout
  DoubleMuParked2012C_10000_Higgs.root:
    type: File
    outputBinding:
       glob: "outputs/DoubleMuParked2012C_10000_Higgs.root"

arguments:
  - prefix: -c
    valueFrom: |
      cp -r ../../code/HiggsExample20112012 .; \
      scram b; \
      cd ../../code/HiggsExample20112012/Level3; \
      mkdir -p ../../../outputs; \
      cmsRun demoanalyzer_cfg_level3data.py
```

```
cwlVersion: v1.0
class: CommandLineTool

baseCommand: /bin/zsh

requirements:
  DockerRequirement:
    dockerPull:
      clelange/cmssw:5_3_32
  InitialWorkDirRequirement:
    listing:
      - $(inputs.code)
      - $(inputs.inputs)

inputs:
  inputs:
    type: Directory
  code:
    type: Directory

stdout: step1mc.log

outputs:
  step1mc.log:
    type: stdout
  Higgs4L1file.root:
    type: File
    outputBinding:
       glob: "outputs/Higgs4L1file.root"

arguments:
  - prefix: -c
    valueFrom: |
      cp -r ../../code/HiggsExample20112012 .; \
      scram b; \
      cd ../../code/HiggsExample20112012/Level3; \
      mkdir -p ../../../outputs; \
      cmsRun demoanalyzer_cfg_level3MC.py
```

The previous workflows calls upon the code which process each of the raw and MC data. Next, both processed data outputs will be run with a plotting code as part of the second step. The workflow for the second step, "*step*2.*cwl*", is described as:

```
cwlVersion: v1.0
class: CommandLineTool

baseCommand: /bin/zsh

requirements:
  DockerRequirement:
    dockerPull:
      clelange/cmssw:5_3_32
  InitialWorkDirRequirement:
    listing:
      - $(inputs.code)
      - $(inputs.inputs)

inputs:
  inputs:
    type: Directory
  code:
    type: Directory
  DoubleMuParked2012C_10000_Higgs:
    type: File
  Higgs4L1file:
    type: File

stdout: step2.log

outputs:
  step2.log:
    type: stdout
  mass4l_combine_userlvl3.pdf:
    type: File
    outputBinding:
      glob: "outputs/mass4l_combine_userlvl3.pdf"

arguments:
  - prefix: -c
    valueFrom: |
      cp -r ../../code/HiggsExample20112012 .; \
      cd ../../code/HiggsExample20112012/Level3; \
      mkdir -p ../../../outputs; \
      cp $(inputs.DoubleMuParked2012C_10000_Higgs.path) ../../../outputs; \
      cp $(inputs.Higgs4L1file.path) ../../../outputs; \
      root -b -l -q ./M4Lnormdatall_lvl3.cc
```

In essence, each step of the analysis has been described separately in individual workflows. This singularity trait gives a certain degree of freedom to the analyses run through ReANA since the workflows steps can be run parallel to one and other and will not depend sequentially to previous workflows. Finally, the last workflow puts together all the commands in terms of workflows and details of the analysis. This closing workflow is named "*workflow*.*cwl*"; as displayed below, the structure of it is in fact elemental to any type of analysis,

```
#!/usr/bin/env cwl-runner

cwlVersion: v1.0
class: Workflow

requirements:
  InitialWorkDirRequirement:
    listing:
      - $(inputs.code)
      - $(inputs.inputs)

inputs:
  inputs:
    type: Directory
  code:
    type: Directory

outputs:
  mass4l_combine_userlvl3.pdf:
    type: File
    outputSource:
      step2/mass4l_combine_userlvl3.pdf

steps:
  step1data:
    run: step1data.cwl
    in:
      code: code
      inputs: inputs
    out: [DoubleMuParked2012C_10000_Higgs.root, step1data.log]
  step1mc:
    run: step1mc.cwl
    in:
      code: code
      inputs: inputs
    out: [Higgs4L1file.root, step1mc.log]
  step2:
    run: step2.cwl
    in:
      code: code
      inputs: inputs
      DoubleMuParked2012C_10000_Higgs: step1data/DoubleMuParked2012C_10000_Higgs.root
      Higgs4L1file: step1mc/Higgs4L1file.root
    out: [mass4l_combine_userlvl3.pdf, step2.log]
```

In order to submit this analysis on the ReANA cloud, a file must be constructed, which recalls the four main characteristics of the analysis (i.e. inputs, code, environment, and workflow), as earlier described. This file follows industry standards originally exposed by the ReANA developer's team. Also the ReANA file, and therefore the cluster itself, understands simple data formats for an easier implementation from the user. This file is used by REANA to instantiate and run the analysis on the cloud. The descriptive ReANA file is standardized for every analysis to be run on ReANA, as so it is named "*reana.yaml*",

```
version: 0.3.0
inputs:
  files:
  - code/HiggsExample20112012/HiggsDemoAnalyzer/src/HiggsDemoAnalyzerGit.cc
  - code/HiggsExample20112012/Level3/demoanalyzer_cfg_level3data.py
  - code/HiggsExample20112012/Level3/demoanalyzer_cfg_level3MC.py
  - code/HiggsExample20112012/Level3/M4Lnormdatall_lvl3.cc
  parameters:
     input: workflow/input.yaml
```

```
workflow:
  type: cwl
  file: workflow/workflow.cwl
outputs:
  files:
   - results/mass4l_combine_userlvl3.pdf
```

In the first stage of the Higgs to 4 leptons example on ReANA, the data analysis job will produce a root output file containing one Higgs candidate from the data, and the simulation analysis job a root file containing the Higgs signal distributions with reduced statistics. At the second stage both files are analyzed using root macro and produce the output plot, displayed below,

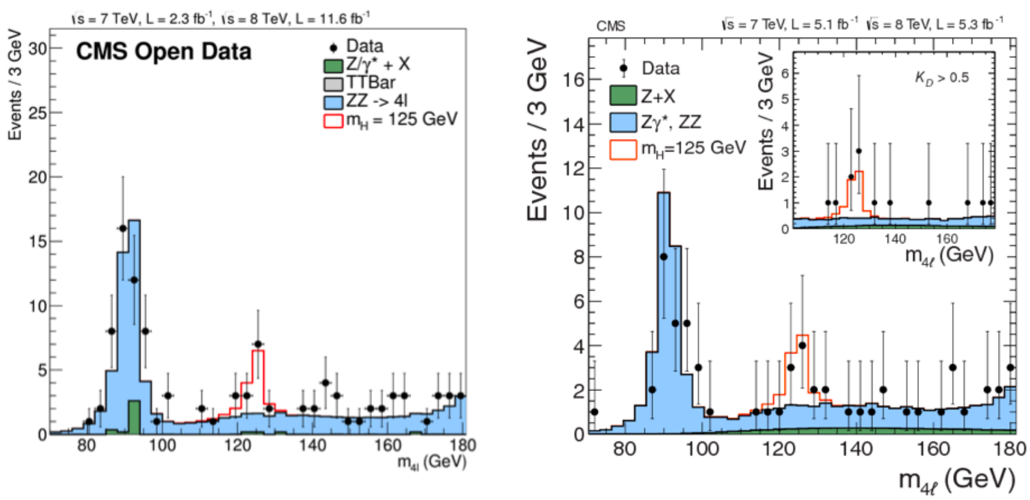

Figure 3: (Left) The output plot from running the analysis on the ReANA cluster using the code from the Open Data Portal. (Right) The resulting plot published at the original paper for the discovery of the Higgs boson.

The output file is compared to the plot from the original publication. Points represent the data, shaded histograms represent the background and unshaded histogram the signal expectations. The expected distributions are presented as stacked histograms. The measurements are presented for the sum of the data collected at $\sqrt{s}$=7 TeV and $\sqrt{s}$=8 TeV. Differences in the data points and margins are most discernible yet both plots presented the expected Higgs mass at 125 GeV. We can appreciate the attributions from Monte Carlo simulated values, already weighted by luminosity, cross-section and number of events, separated by ZZ, a pair of heavier bosons, TTBar, a pair of top and anti-top quarks, and some irreducible background from singular Z bosons. At the end, adding these contributions plus the data result in a distribution of the four-lepton reconstructed mass in the low mass region for the sum of the 4e, 4μ and $2e2\mu$ channels.

3.1.1 Running the analysis in the ReANA platform

An analysis can be made reproducible, i.e. rerunnable, by defining the inputs taken into account, the stages which organize its components, the current compute environment and costumed workflow files. It is implied that the contributor has the code locally in their computer, and has a broad idea of which tools are needed like specific analysis frameworks, custom analysis code, Jupyter notebooks, etc. , and how to run it successfully. Encapsulating the environment involves archiving the type of operating systems, software packages and libraries, CPU and memory resources, among other technicalities. Moreover, the workflow files would describe the steps taken to execute the analysis, if they were simple shell command or complex computational workflows, in order to get expected outcomes such as plots, histograms, and any other files. After structuring the analysis, the research repository should be organized according to inputs, code, environments, and workflows directories.

Next step is, ideally, to install the reana-client. This process uses the LXPLUS Cluster, lxplus.cern.ch, for which the user simply logs in using their account. The initial preparation to run analyses on the ReANA cloud is to install, in a new virtual environment, the latest package for the reana-client. This process can be done by executing the following commands:

```
$ # install REANA client:
$ mkvirtualenv reana-client
$ pip install reana-client
$ # connect to some REANA cloud instance:
$ export REANA_SERVER_URL=https://reana.cern.ch/
$ export REANA_ACCESS_TOKEN=XXXXXXX
$ # create new workflow:
$ reana-client create -n my-analysis
$ export REANA_WORKON=my-analysis
$ # upload input code and data to the workspace:
$ reana-client upload ./code ./data
$ # start computational workflow:
$ reana-client start
$ # ... should be finished in about a minute:
$ reana-client status
$ # list workspace files:
$ reana-client list
$ # download output results:
$ reana-client download results/mass4l_combine_userlvl3.pdf
```

Additionally to the installing, the user must connect to some ReANA cloud instance for which a unique token serves as an identifier while using the ReANA client. Lastly, the creation of workflows and start of the analysis process are shown above.

3.2 Reprocessing AOD from 2010-2012 RAW samples

In addition to the analysis examples, such as the Higgs to 4 lepton analysis described above, CMS also provides code for validation of the datasets served through the CERN Open Data Portal. The inputs are datasets specific to every example, and the code for the analysis used with the CMS Open Data Virtual Machine environment. Validation examples prove that a result can be obtained with the legacy or open data tools. For instance, a validation example for 2010 datasets consists of: setting up the CMS environment, compiling and running the analysis, and the workflows written for producing a result (e.g. a comparison plot for the dimuon mass spectrum). The ReANA setup described above allows to rerun CMS Open Data analyses and in doing so it confirms that instructions are correct and runs large-scale analyses relatively fast.

Further testing of the preservation and reproducibility platforms is in order, therefore, for the newest data release from the Open Data Portal there is an example that entails the reconstruction of open data and comparison of such. This example is a simple comparison to validate the reprocessing step on the Open Data VM (Virtual Machine) by comparing the resulting file, newly reconstructed from the RAW data samples from 2010 to 2012, and the original AOD available on the Open Data Portal. The selected data samples for the analysis, and to ones which will be published in the next CMS Open Data release in January 2019, are:

/MinimumBias/Run2010B-v1/RAW
/Electron/Run2010B-v1/RAW
/Mu /Run2010B-v1/RAW
/Jet /Run2010B-v1/RAW
/DoubleElectron/Run2011A-v1/RAW
/SingleElectron/Run2011A-v1/RAW
/DoubleMu/Run2011A-v1/RAW
/SingleMu/Run2011A-v1/RAW
/Jet/Run2011A-v1/RAW
/DoubleElectron/Run2012B-v1/RAW
/SingleElectron/Run2012B-v1/RAW
/DoubleMuParked/Run2012B-v1/RAW
/SingleMu/Run2012B-v1/RAW
/JetHT/Run2012B-v1/RAW

The analysis workflow for this example involves a step for the data reconstruction and another for a basic analyzer plotter. The process of data reconstruction requires a configurations file for each of the year of data, in this case 2010, 2011, and 2012. Moreover, the configuration file was created using a CMS tool embedded in a CMSSW environment, cmsDriver.py. The configuration code takes as input one file from the selected datasets, chosen at random from the EOS Public Browser [19]. Additional changes to the file, including the number of maximum events to be processed, and GlobalTag (i.e. a record of conditions used in the CMS data processing workflows specific for every year of data taking), were made to customize the reconstruction process accordingly. For the creation of histograms, a validation code to plot basic physics objects from AOD [20] was taken as a skeleton code to further validate the data reconstruction results. The plotting code loops over different physics objects, such as tracks, electrons, muons, photons, jets, taus and missing et, and fills histograms with P, pt, eta and phi of these objects. These measurements recall to momentum, transverse momentum, spatial coordinate describing the angle of a particle relative to the beam axis, and polar angle in the transverse plane.

The newly reprocessed AOD from RAW is compared to the results given from the corresponding AOD files from the Open Data Portal. To compare these results, it was taken in consideration the histograms showing the electron momentum for all the samples tested:

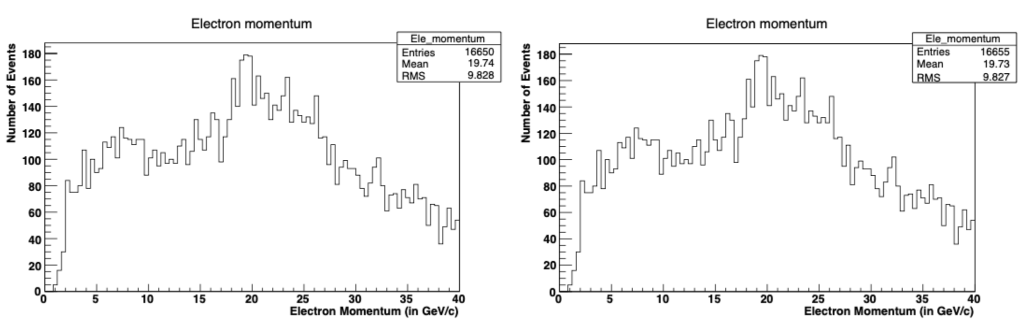

Figure 4: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /Electron/Run2010B-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

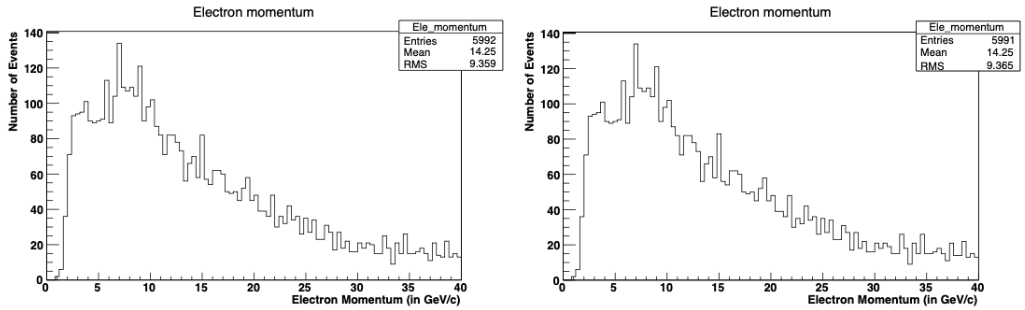

Figure 5: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /Jet/Run2010B-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

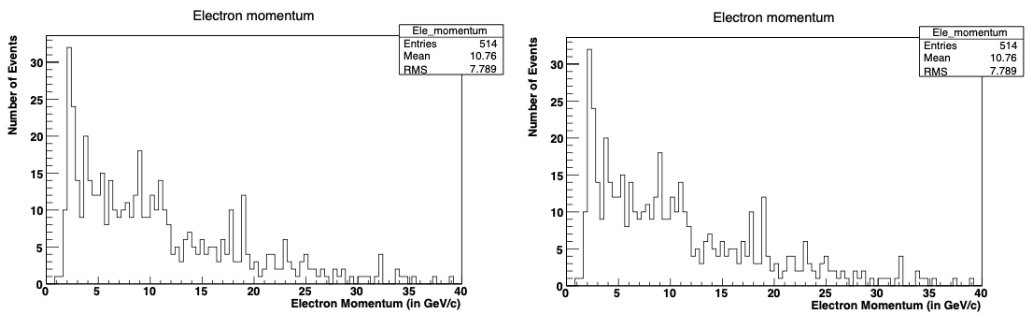

Figure 6: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /MinimumBias/Run2010B-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

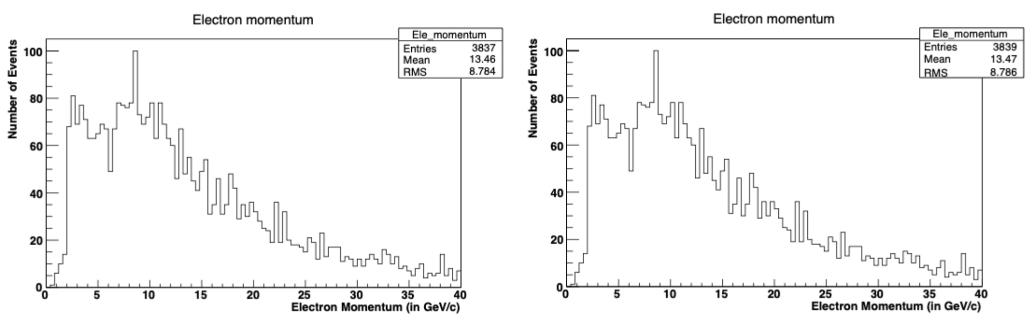

Figure 7: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /Mu/Run2010B-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

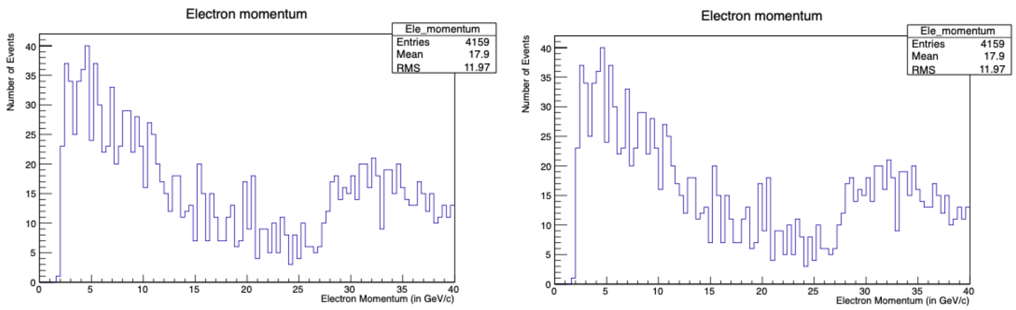

Figure 8: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /SingleElectron/Run2011A-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

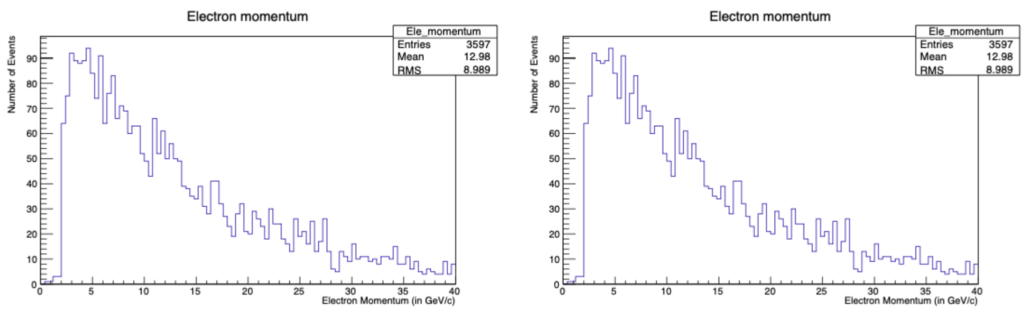

Figure 9: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /SingleMu/Run2011A-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

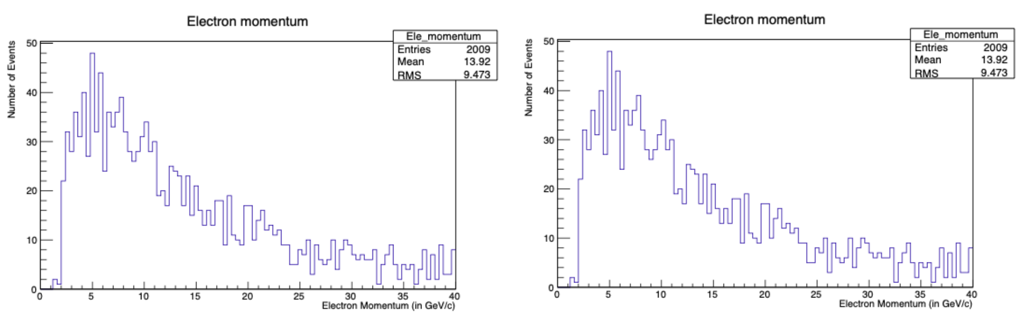

Figure 10: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /Jet/Run2011A-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

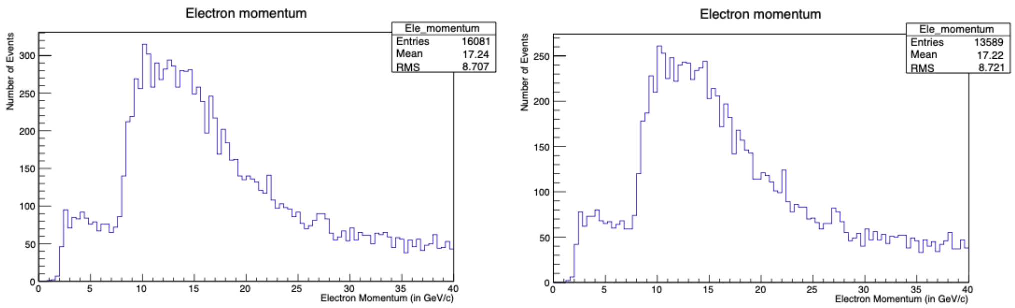

Figure 11: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /DoubleElectron/Run2011A-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

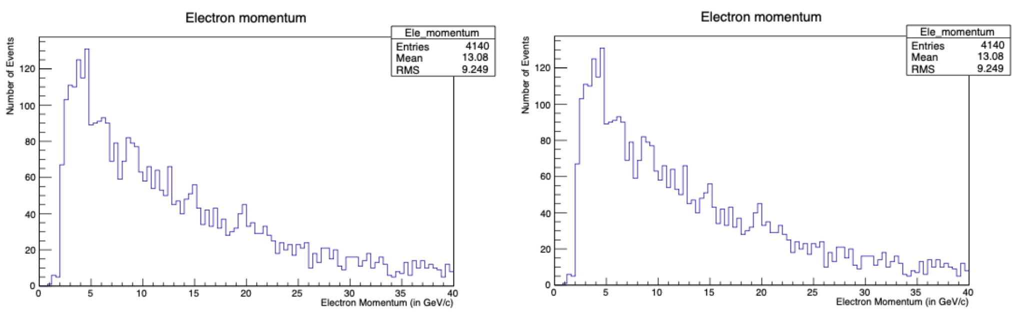

Figure 12: (Left) Resulting histogram of electron momentum using reconstructed AOD file from /DoubleMu/Run2011A-v1/RAW sample. (Right) Output histogram of electron momentum from the Open Data AOD corresponfing file.

All RAW samples were reconstructed successfully, and have one-to-one match with the original AOD. To run this process through ReANA, CWL written workflow are created for each of the steps. The analysis needs cvmfs (CERN Virtual Machine File System) access in order to read condition data details in the analysis, meaning both a CWL tool and a Docker container must have these CMSSW attributes in order to successfully be reproducible.

4 Conclusions

In this work, CMS analysis examples from CERN Open Data Portal are taken as examples to connect the three data preservation services, CODP, CAP and ReANA. The preservation of these physics analysis examples ensures that the complete information is structured and reproducible. Workflow steps for the examples were tested locally and on a remote server, proving to be successful in producing the expected outputs. In ReANA, access to the experimental datasets, analysis software area, operating system environment and computational workflow steps remains to be tested. Overall, the implementation of preservation and reproducibility through the CERN Analysis Preservation and ReANA looks promising as more examples continue to be evaluated.

References

[1] CMS Collaboration, G.L. Bayatyan et al. CMS Technical Proposal. CERN-LHCC-94-38, 1994.

[2] L. Evans and P. Bryant. LHC Machine. JINST, 3, 2008.

[3] CERN. 2018, https://home.cern.

[4] The CMS Collaboration. The CMS experiment at the CERN LHC. JINST, 3, 2008.

[5] CMS Collaboration 2009 Particle–Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET CMS-PAS-PFT-09-001

[6] CMS Collaboration 2010 Commissioning of the Particle-Flow Reconstruction in Minimum-Bias and Jet Events from pp Collisions at 7 TeV CMS-PAS-PFT-10-002

[7] CERN Open Data Portal. 2018, http://opendata.cern.ch.

[8] CERN Analysis Preservation Portal. 2018, https://analysispreservation.cern.ch.

[9] ReANA: Reusable Analyses. 2018, http://www.reana.io/.

[10] Jomhari, Nur Zulaiha; Geiser, Achim; Bin Anuar, Afiq Aizuddin. Higgs-to-four-lepton analysis example using 2011-2012 data. CERN Open Data Portal. 2017, DOI:10.7483/OPENDATA.CMS.JKB8.RR42.

[11] Glashow S. L. Partial Symmetries of Weak Interactions Nucl. Phys., 1961.

[12] Weinberg S. A Model of Leptons Phys. Rev. Lett., 1967.

[13] Salam A 1968 Elementary Particle Physics N. Svartholm ed., Almquvist and Wiksell, Stockholm

[14] CERN IT Department. LXPLUS Service. 2013, http://information-technology.web.cern.ch/services/lxplus-service

[15] CMS Collaboration. "CMS VM Image, for 2011 and 2012 CMS open data". 2014, http://opendata.cern.ch/record/252.

[16] YAML Ain't Markup Language. 2018, http://yaml.org

[17] "reanahub/reana-demo-cms-h4l". Github, 2018, https://github.com/reanahub/reana-demo-cms-h4l.

[18] Common Workflow Language. 2018, https://www.commonwl.org/.

[19] EOS HTTP Browser. 2018, https://eospublichttp01.cern.ch/eos/opendata/cms.

[20] Lassila-Perini, Kati. Validation code to plot basic physics objects from AOD. CERN Open Data Portal. 2017, DOI:10.7483/OPENDATA.CMS.11RI.SDX7