Common Analysis Tools in CMS

Tommaso Tedeschi[1] on behalf of the CMS Collaboration

[1]Istituto Nazionale di Fisica Nucleare - Sezione di Perugia, Italy

E-mail: tommaso.tedeschi@pg.infn.it

Abstract. The CMS experiment has recently established a new Common Analysis Tools (CAT) group. The CAT group implements a forum for the discussion, dissemination, organization and development of analysis tools, broadly bridging the gap between the CMS data and simulation datasets and the publication-grade plots and results. In this contribution, we discuss some of the recent developments carried out in the group, including its structure, facilities and services provided, communication channels, ongoing developments in the context of frameworks for data processing, strategies for the management of analysis workflows and their preservation, and tools for the statistical interpretation of analysis results.

1 Introduction

The Compact Muon Solenoid (CMS) [1] is one of the four main experiments at the Large Hadron Collider (LHC, [2]), the largest and most powerful particle accelerator in the world. It is a general-purpose apparatus for investigating a wide range of high energy physics (HEP) processes. The CMS Collaboration consists of over 4000 particle physicists, engineers, computer scientists, technicians and students from around 240 institutes and universities from more than 50 countries. Since the very beginning of CMS operations, data collected (or simulated) by the experiment has been stored into ROOT [3] files, resulting from the reconstruction in the CMS software (commonly referred to as CMSSW), hosted in an open-source repository [4]. The CMSSW software stack (mostly C++ and python code) includes particle generators, high-level trigger and low-level trigger emulation code, offline workflows for data and simulation processing, and some analysis code. In order to constantly match community needs, different data formats (also known as *datatiers*) were introduced to convey full or slimmed event information reconstructed directly from either the detector readout or its simulation (known as RAW, whose size is of the order of 1MB per event, and whose full reconstruction corresponds to the RECO format). In particular, the following slimmed formats were progressively introduced:

- AOD (acronym of Analysis Object Data), which was introduced in 2011, with almost half the size of the RAW format;
- MiniAOD [5], which was introduced in 2013, reducing by an order of magnitude the size of AOD;
- NanoAOD [6], which was introduced in 2018 and is about an order of magnitude smaller than MiniAOD: the key to achieving this reduction is the usage of basic data types (e.g. float, int, and arrays thereofs) and plain ROOT TTrees, storing just variables related to high-level physical objects, including pre-calculated quantities related to their identification (filtered using appropriate thresholds).

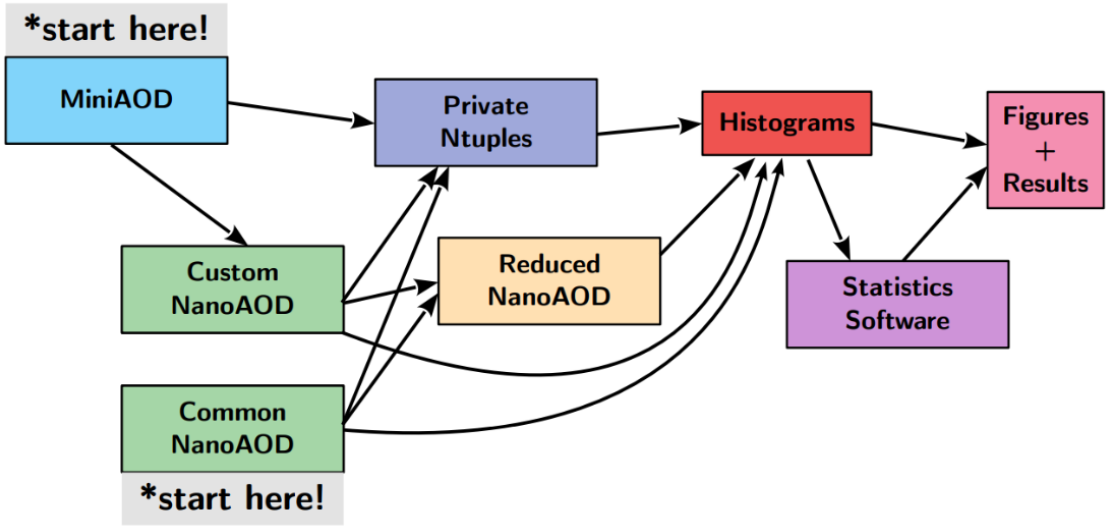

Figure 1: Possible different workflows (excluding special cases) of the hundreds of ongoing CMS Run2 and Run3 analyses.

Starting from those *datatiers*, CMS physicists develop their analysis: as can be seen in Fig. 1, most of the Run2 and Run3 analyses start from MiniAOD or NanoAOD (summing up to tens or hundreds of TBs of real and simulated events), and usually go through one or more intermediate reduction or augmentation steps (exploiting HTC clusters on the grid [7], at CERN, or at home institutions) which, with a typical turnaround of hours or days, produce final data-reduced quantities (histograms, predominantly). The latter are typically then fed to statistical inference software and used to produce the final results.

Different tools are required to accomplish each of the steps above. These are often custom or customized and may vary across different analysis groups. Given the lively development rate and the high heterogeneity of end-user data analysis tools, CMS has recently established a Common Analysis Tools (CAT) group to provide a forum where the discussion of end-user analysis tools could happen, identify and prevent further duplication of work and find viable support patterns. Some of the main responsibilities of the new group are the provisioning of centralized support and documentation for a selected subset of tools of common interest, with a keen eye on efficiency, interactivity, and re-usability of analysis code. The activity of this newborn group is described in the next section.

2 Common Analysis Tools

The CMS Common Analysis Tools group (CAT), [8]) was established in September 2022 and is charged with two main tasks: taking ownership of the development, maintenance and documentation of analysis tools of common interest and providing a forum to discuss developments of new analysis tools, offering guidance. In order to achieve this ambitious goal, the CAT group needs a complex internal organization and several discussion venues to develop and support all the different aspects of data analysis. All of this will be detailed in the following.

2.1 Organization

The CAT group is led by two conveners and is organized into three subgroups, each coordinated by two sub-conveners. These are:

- Data Processing Tools (DPROC) subgroup, which has the responsibility for the support, management, and development of tools running directly on the CMS centrally-produced datasets;
- Workflow Orchestration and Analysis Preservation (WFLOWS) subgroup, charged with the support, management, and development of tools for the configuration, coordination, and management of physics analysis workflows, promoting tools that ease the long-term reproducibility of analyses;
- Statistical Interpretation Tools (STATS) subgroup, which works on the support, management and development of statistical interpretation tools (with a focus on the `Combine` [9] tool).

CAT-related discussions, developments and dissemination happen at several venues. General meetings are held typically every two weeks, where news and contributions on recent developments are reported, with dedicated slots for introducing new work. The main communication channels are CMS-talk (a customized version of Discourse [10]) and the dedicated CAT documentation website. The latter includes recommendations for CMS analysis and instructions on how to setup analysis code areas, an overview of supported tools for data processing, workflow management and statistical analysis, useful snippets, a collection of links to Collaboration-wide accessible Analysis Facilities, tutorials, communication channels and other guidelines. Documentation pages are built using `mkDocs` [11], which renders markdown source hosted on `gitlab.cern.ch`.

To promote participation, the CAT group organizes periodical HaCAThons (mixed hacking and training events), which gather dozens of members of the Collaboration willing to contribute on different topics of common interest.

2.2 Unified analysis code area

With the goal of reusability, reproducibility and preservation of analysis in mind, the CAT group maintains a unified code area for analyses (schematically depicted in Fig. 2): each analysis group is encouraged to put their analysis code (or at least mirror it) in this area. All `Combine` input files are mandated to be hosted in the common area. With newer analysis tools, user analysis code is represented by just a configuration layer (implemented with one or more files) on top of a common framework, whose core code is hosted in the same `GitLab` group too. In addition, CAT encourages and provides training and templates within the common area for the integration of analysis code with CI/CD, implementing code checks and automatic versioned container images building.

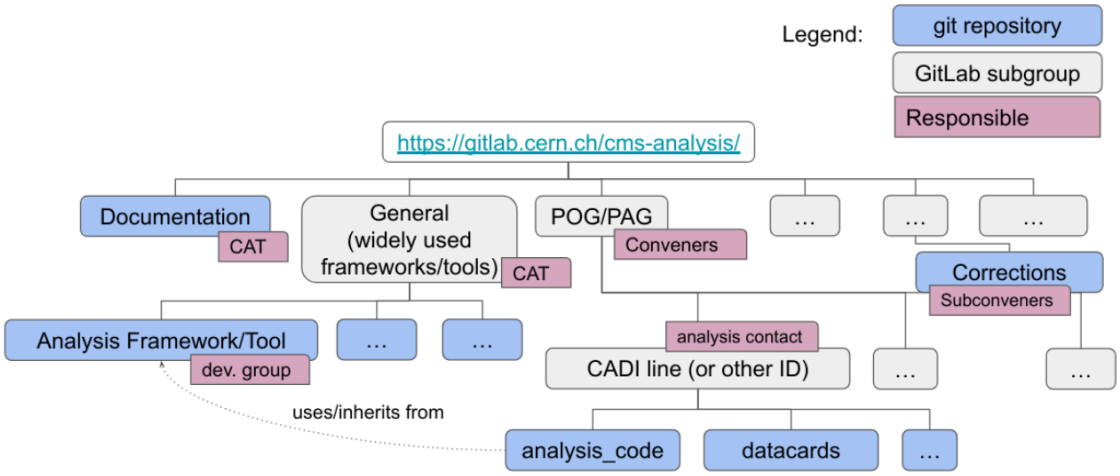

Figure 2: `cms-analysis` area structure, which hosts analysis tools core code, analysis code and POG/PAG-specific code (the CMS Physics Coordination area has 8 groups dedicated to different thematic physics analyses, PAGs, and 8 groups focused on physics objects, POGs). Taken from [8].

2.3 Supported analysis tools

As part of its mandate, the CAT group is progressively collecting a series of so-called CAT-supported tools (also referred to as frameworks). These consist of CMS-specific analysis tools (of common interest) that meet some specific requirements, such as residing or being mirrored in the aforementioned code area or CMSSW, and being actively developed, documented, maintained and supported by identified teams. A dedicated page in CAT documentation describes their functionalities and points to relevant documentation.

The aforementioned analysis frameworks include tools for both physical objects studies and end-user analysis, characterized by declarative approaches, efficiency and (quasi-)interactivity, which leads to a reduction of time-to-insight. Therefore, those frameworks are mostly based on the emerging next-gen data processing tools for HEP, i.e. ROOT's RDataFrame (RDF) [12] and HSF's Awkward Arrays/Coffea

[13], and target the NanoAOD format as preferred *datatier* due to its flatness and lightness. Table 1 reports all the current CAT-supported analysis frameworks.

Table 1: List of CAT-supported analysis frameworks

Framework	Description
`nanoAOD-tools` [14]	legacy pyROOT-based sequential framework to skim/extend nanoAODs, and produce plots
`bamboo` [15]	RDataFrame-based python framework that allows to express analysis in a functional style
`CROWN` [16]	RDataFrame-based (C++ and python) framework to generate analysis ntuples (and friends)
`columnflow` [17]	python (Awkward Arrays)-based backend for columnar, fully-orchestrated HEP analyses
`DasAnalysisSystem` [18]	ROOT-based tools for analysis with high-level objects
`PocketCoffea` [19]	configuration framework for Coffea-based analyses on NanoAODs
`mkShapesRDF` [20]	RDataFrame-based framework for analyses on NanoAODs, which are implemented through configuration files

CAT also contributed to the recent update of `mplhep` [21] and to the introduction of the new `cmsstyle` [22] package, which allow CMS users to easily produce production-ready plots in python: in particular, `mplhep` relies on the `scikit-hep` [23] ecosystem (being it an extension of `matplotlib`) while `cmsstyle` depends on ROOT python bindings (pyROOT). CMS default color-vision-deficiency friendly color schemes (recently voted by the Collaboration) are used as defaults in both tools, as shown in Fig. 3.

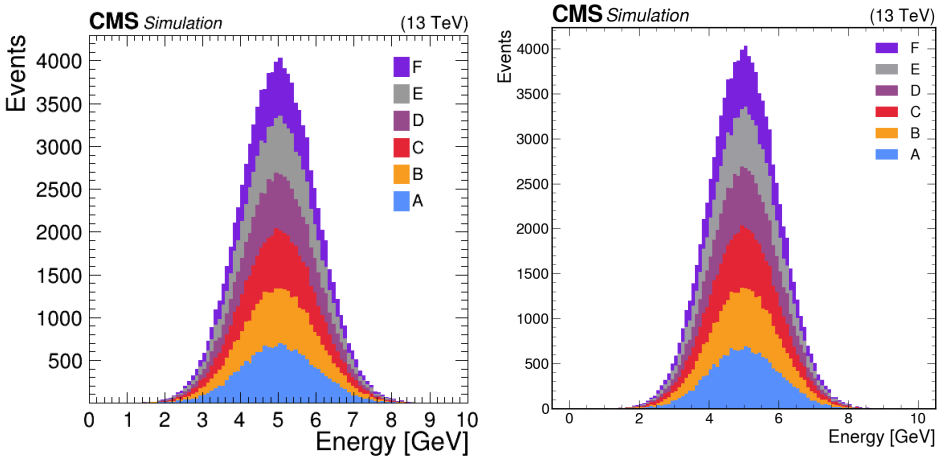

Figure 3: Example 1D histograms created starting from the same randomly-generated Gaussian distributions, via `cmsstyle` library (left) and `mplhep` library (right). The color scheme automatically implemented is the one voted by the CMS Collaboration, and corresponds to the one suggested in [24]. Images are taken from [8].

2.4 Metadata management

Another key aspect in CMS data analysis is the metadata handling, i.e. all information regarding collected data and simulated samples that are necessary to correctly extract physical results along with

their uncertainties. In this respect, the CAT group aims at a fully automated and versioned way to access that information.

On the one hand, this includes a work to enable the distribution of analysis metadata via `/cvmfs` (see Fig. 4 for the designed schema).

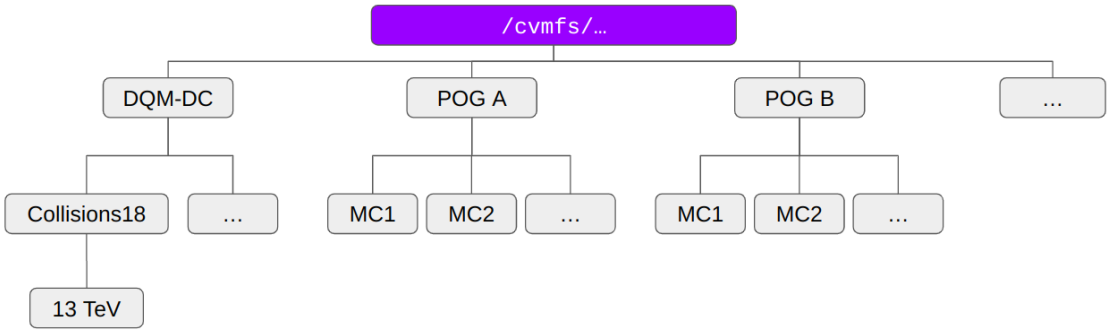

Figure 4: Sketch representing proposed metadata distribution via `/cvmfs`. Taken from [8].

On the other hand, this also requires a tool capable of retrieving this information and building sophisticated structures to handle it all at once. The `order` [25] tool is being implemented with this functionality in mind.

2.5 Statistical tools

As mentioned in the beginning, the CAT group is also charged with contributing to the support of statistical tools of common interest for the Collaboration, and most importantly of `Combine` [9], which is a `RooStats`/`RooFit`-based software tool that represents the *de-facto* standard for statistical analysis within the CMS experiment. It provides a command-line interface to many different statistical techniques and statistical models are encapsulated using a human-readable configuration file (commonly referred to as *datacard*).

In addition to that, CAT contributes to the discussion on a HEP-wide standard for the description of likelihoods in collaboration with other experiments, as well as other developments related to the `Combine` ecosystem.

3 Summary

This work presented some of the achievements of CAT group in 1.5 years of operations. While it is true that much progress has been done in all steps of data analysis, moving towards efficient, reproducible, and easy ways of doing analysis, it is also true that still much work is to do in various directions, including moving further towards automation, containerization and unification of metadata, in addition to the maintenance of what has been already done.

Acknowledgements

T. Tedeschi would like to thank the whole CMS Collaboration, in particular the Common Analysis Tools Group conveners and contributors, and Evan Altair Ranken for providing the first figure. T. Tedeschi acknowledges support from the ICSC – National Research Center for High Performance Computing, Big Data and Quantum Computing, funded by the NextGenerationEU program (Italy).

References

[1] The CMS Collaboration. The CMS experiment at the CERN LHC. *Journal of Instrumentation*, 3:S08004, 2008.

[2] Thomas Sven Pettersson and P Lefèvre. The Large Hadron Collider: conceptual design. Technical report, 1995.

[3] Rene Brun and Fons Rademakers. ROOT — An object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1):81–86, 1997. New Computing Techniques in Physics Research V.

[4] CMS Collaboration. CMSSW. https://github.com/cms-sw/cmssw, 2024. [Accessed: 15/07/2024].

[5] Giovanni Petrucciani, Andrea Rizzi, and Carl Vuosalo. Mini-AOD: A New Analysis Data Format for CMS. *Journal of Physics: Conference Series*, 664(7):072052, 2015.

[6] Marco Peruzzi, Giovanni Petrucciani, and Andrea Rizzi. The NanoAOD event data format in CMS. *Journal of Physics: Conference Series*, 1525(1):012038, 2020.

[7] Jamie Shiers. The Worldwide LHC Computing Grid (worldwide LCG). *Computer Physics Communications*, 177(1):219–223, 2007. Proceedings of the Conference on Computational Physics 2006.

[8] CMS Collaboration. Recent developments of the CMS Common Analysis Tools group in Data Processing, Workflow Orchestration and Analysis Preservation. https://cds.cern.ch/record/2902863, 2024.

[9] CMS Collaboration. The CMS statistical analysis and combination tool: COMBINE. Submitted to *Comput. Softw. Big Sci.*, 2024.

[10] Civilized Discourse Construction Kit Inc. Discourse. https://www.discourse.org/, 2024. [Accessed: 15/07/2024].

[11] Tom Christie. MkDocs,Project documentation with Markdown. https://www.mkdocs.org/, 2024. [Accessed: 15/07/2024].

[12] Danilo Piparo, Philippe Canal, Enrico Guiraud, Xavier Valls Pla, Gerardo Ganis, Guilherme Amadio, Axel Naumann, and Enric Tejedor Saavedra. RDataFrame: Easy parallel ROOT analysis at 100 threads. *EPJ Web Conf.*, 214:06029, 2019.

[13] Nicholas Smith, Lindsey Gray, Matteo Cremonesi, Bo Jayatilaka, Oliver Gutsche, Allison Hall, Kevin Pedro, Maria Acosta, Andrew Melo, Stefano Belforte, and Jim Pivarski. Coffea columnar object framework for effective analysis. *EPJ Web Conf.*, 245:06012, 2020.

[14] cms-sw. NanoAOD-tools, 2024. https://github.com/cms-sw/cmssw/tree/master/PhysicsTools/NanoAODTools [Accessed: 15/07/2024].

[15] cms-analysis. bamboo, 2024. https://gitlab.cern.ch/cms-analysis/general/bamboo [Accessed: 15/07/2024].

[16] cms-analysis. CROWN, 2024. https://gitlab.cern.ch/cms-analysis/general/crown [Accessed: 15/07/2024].

[17] cms-analysis. columnflow, 2024. https://gitlab.cern.ch/cms-analysis/general/columnflow [Accessed: 15/07/2024].

[18] cms-analysis. DasAnalysisSystem, 2024. https://gitlab.cern.ch/cms-analysis/general/DasAnalysisSystem [Accessed: 15/07/2024].

[19] cms-analysis. PocketCoffea, 2024. https://gitlab.cern.ch/cms-analysis/general/PocketCoffea [Accessed: 15/07/2024].

[20] cms-analysis. mkShapesRDF, 2024. https://gitlab.cern.ch/cms-analysis/general/mkShapesRDF [Accessed: 15/07/2024].

[21] scikit-hep. mplhep, 2024. https://github.com/scikit-hep/mplhep [Accessed: 15/07/2024].

[22] cms-cat. cmsstyle, 2024. https://github.com/cms-cat/cmsstyle [Accessed: 15/07/2024].

[23] Eduardo Rodrigues, Benjamin Krikler, Chris Burr, Dmitri Smirnov, Hans Dembinski, Henry Schreiner, Jaydeep Nandi, Jim Pivarski, Matthew Feickert, Matthieu Marinangeli, Nick Smith, and Pratyush Das. The Scikit HEP Project overview and prospects. *EPJ Web Conf.*, 245:06028, 2020.

[24] Matthew A. Petroff. Accessible color cycles for data visualization. *CoRR*, abs/2107.02270, 2021.

[25] Marcel Rieger. order. https://github.com/riga/order, 2024. [Accessed: 15/07/2024].