

# ROOT Users Workshop 2025

Monday 17 November 2025 - Friday 21 November 2025



## Book of Abstracts



# Contents

Welcome from the organisers of the workshop . . . . .	1
Welcome from the local organisers . . . . .	1
An overview of the ROOT project . . . . .	1
ALICE . . . . .	1
ATLAS . . . . .	1
CMS . . . . .	1
Talk 4 . . . . .	1
ROOT Training Session . . . . .	1
Talk 1 . . . . .	1
Talk 2 . . . . .	2
Talk 3 . . . . .	2
Talk 4 . . . . .	2
Discussion . . . . .	2
WLCG . . . . .	2
”root://”: Current state and future challenges of Physics data storage at scale . . . . .	2
Plenary 3 . . . . .	2
Overview of the current RDataFrame efforts . . . . .	2
Talk 2 . . . . .	3
Discussion . . . . .	3
Talk 3 . . . . .	3
Talk 4 . . . . .	3
Discussion . . . . .	3
Talk 1 . . . . .	3

Talk 2 . . . . .	3
Discussion . . . . .	3
Talk 3 . . . . .	4
Talk 4 . . . . .	4
Discussion . . . . .	4
Future of RDataFrame - R&D projects ideas . . . . .	4
Talk 2 . . . . .	4
Discussion . . . . .	4
Talk 3 . . . . .	4
Talk 4 . . . . .	4
Discussion . . . . .	4
Talk 1 . . . . .	5
Talk 2 . . . . .	5
Discussion . . . . .	5
Talk 3 . . . . .	5
Talk 4 . . . . .	5
Discussion . . . . .	5
Talk 1 . . . . .	5
Talk 2 . . . . .	5
Discussion . . . . .	6
Talk 3 . . . . .	6
Talk 4 . . . . .	6
Discussion . . . . .	6
Talk 1 . . . . .	6
Talk 2 . . . . .	6
Discussion . . . . .	6
Talk 3 . . . . .	6
Talk 4 . . . . .	6
Discussion . . . . .	7
Plenary 1 . . . . .	7

Plenary 2 . . . . .	7
Plenary 3 . . . . .	7
Plenary 1 . . . . .	7
Plenary 2 . . . . .	7
Plenary 3 . . . . .	7
Talk 1 . . . . .	7
Talk 2 . . . . .	8
Discussion . . . . .	8
Talk 3 . . . . .	8
Talk 4 . . . . .	8
Discussion . . . . .	8
Talk 1 . . . . .	8
Talk 2 . . . . .	8
Discussion . . . . .	8
Talk 3 . . . . .	8
Talk 4 . . . . .	9
Discussion . . . . .	9
TDAnalyser - a modular framework for the analysis of test beam data . . . . .	9
Bridging Gravitational Wave and High-Energy Physics Software . . . . .	9
The outside view on ROOT . . . . .	10
Performance Comparison of Lossless Compression Algorithms on CMS Data using ROOT TTree and RNTuple . . . . .	10
Vibe Plotting: ServiceX, RDataFrame, and a LLM . . . . .	11
RNTuple for SND@HL-LHC . . . . .	11
Designing and evolving a (Py)ROOT based software framework for SHiP . . . . .	11
The Utilization of ROOT in the BESIII Offline Software System . . . . .	12
CMS FlashSim and ROOT RDataFrame for ML-Based Event Simulation . . . . .	12
Implementation of an $H \rightarrow \tau\tau$ Analysis within a ROOT-based C++ Framework using ATLAS Open Data . . . . .	13
Introduction to the JUNO offline data processing software and its event data model . . .	13

NDMSPC: Addressing THnSparse Challenges in High-Dimensional Analysis . . . . .	14
NDimensional Visualization in 3D Virtual Environments . . . . .	14
User Story: Integration of ROOT RNTuple to CMSSW's SoA data structures . . . . .	14
Market Surveillance at Scale: A Deployed ROOT Framework for Financial Integrity . . .	15
Beyond HEP: ROOT for Solar Resources . . . . .	15
FCCAnalyses: a ROOT-based Framework for End-to-End Physics Analysis at the FCC . .	15
ROOT.jl, opening ROOT to the Julia programming language . . . . .	15
Cross compilation of the ROOT libraries . . . . .	16
ROOT in the LHCb Software Framework - past, present and future . . . . .	16
Usage of EDM4hep datamodel in RDataFrame . . . . .	16
Benchmarking ROOT-Based Analysis Workflows for HL-LHC: The CMS INFN AF Use Case . . . . .	17
Highlights of xRooFit - The High-Level API for RooFit . . . . .	17
Cloud-Based Machine Learning with ROOT for the High-Luminosity LHC Era . . . . .	18
Efficient Parameter Inference with MoreFit . . . . .	18
High performance analysis in CMS . . . . .	18
Analyzing HL-LHC heavy-ion collision data with ALICE . . . . .	19
The View from Deep Underground: DUNE's Perspective on ROOT . . . . .	19
FORM: Flexible I/O and Storage for the Deep Underground Neutrino Experiment . . . .	20
JSRoot in Web Applications for High-Energy Physics: From Interactive Visualizations to Particle Transport Simulations . . . . .	20
The FairRoot framework . . . . .	21
The R3BRoot framework . . . . .	21
Physics at the HL-LHC scale . . . . .	22
An overview of ROOT in Python . . . . .	22
The ROOT Python bindings features . . . . .	22
Compiler Research (provisional title) . . . . .	22
Discussion . . . . .	22
Innovative Services for Federated Infrastructures - Deployment and Execution in the Com- puting Continuum . . . . .	22
RooFit overview . . . . .	22

A look at statistical inference in HEP (Provisional title) . . . . .	23
CMS Combine (provisional title) . . . . .	23
ATLAS statistical inference (provisional title) . . . . .	23
ROOT software training . . . . .	23
ROOT I/O Status Overview . . . . .	23
A first look at RFile . . . . .	23
RNTuple advanced topics . . . . .	23
Future of ROOT I/O . . . . .	23
The software stack for generic dynamic Python bindings . . . . .	24
Hands-on Pythonisations tutorial . . . . .	24
Workshop discussions and closing . . . . .	24
Workshop registration . . . . .	24
Computing in the 1990s . . . . .	24
30 years of ROOT . . . . .	24





**Morning Session I / 1**

**Welcome from the organisers of the workshop**

**Morning Session I / 2**

**Welcome from the local organisers**

**Morning Session I / 3**

**An overview of the ROOT project**

**Morning Session II / 4**

**ALICE**

**Morning Session II / 5**

**ATLAS**

**Morning Session II / 6**

**CMS**

**Morning Session II / 7**

**Talk 4**

**Afternoon session I / 8**

**ROOT Training Session**

**Afternoon Session II / 9**

## **Talk 1**

**Afternoon Session II / 10**

## **Talk 2**

**Afternoon Session II / 11**

## **Talk 3**

**Afternoon Session II / 12**

## **Talk 4**

**Afternoon Session II / 13**

## **Discussion**

**Morning Session I / 14**

## **WLCG**

**Morning Session I / 15**

**”root://”: Current state and future challenges of Physics data storage at scale**

**Morning Session I / 16**

## **Plenary 3**

**Morning Session II / 17**

## **Overview of the current RDataFrame efforts**

**Morning Session II / 18**

### **Talk 2**

**Morning Session II / 19**

### **Discussion**

**Morning Session II / 20**

### **Talk 3**

**Morning Session II / 21**

### **Talk 4**

**Morning Session II / 22**

### **Discussion**

**Afternoon session I / 23**

### **Talk 1**

**Afternoon session I / 24**

### **Talk 2**

**Afternoon session I / 25**

### **Discussion**

**Afternoon session I / 26**

## **Talk 3**

**Afternoon session I / 27**

## **Talk 4**

**Afternoon session I / 28**

## **Discussion**

**Morning Session II / 29**

## **Future of RDataFrame - R&D projects ideas**

**Morning Session II / 30**

## **Talk 2**

**Morning Session II / 31**

## **Discussion**

**Morning Session II / 32**

## **Talk 3**

**Morning Session II / 33**

## **Talk 4**

**Morning Session II / 34**

## **Discussion**

**Afternoon session I / 35**

### **Talk 1**

**Afternoon session I / 36**

### **Talk 2**

**Afternoon session I / 37**

## **Discussion**

**Afternoon session I / 38**

### **Talk 3**

**Afternoon session I / 39**

### **Talk 4**

**Afternoon session I / 40**

## **Discussion**

**Morning Session II / 41**

### **Talk 1**

**Morning Session II / 42**

### **Talk 2**

**Morning Session II / 43**

## **Discussion**

**Morning Session II / 44**

## **Talk 3**

**Morning Session II / 45**

## **Talk 4**

**Morning Session II / 46**

## **Discussion**

**Afternoon session I / 47**

## **Talk 1**

**Afternoon session I / 48**

## **Talk 2**

**Afternoon session I / 49**

## **Discussion**

**Afternoon session I / 50**

## **Talk 3**

**Afternoon session I / 51**

## **Talk 4**

**Afternoon session I / 52**

## **Discussion**

**Morning Session I / 53**

## **Plenary 1**

**Morning Session I / 54**

## **Plenary 2**

**Morning Session I / 55**

## **Plenary 3**

**Morning Session I / 56**

## **Plenary 1**

**Morning Session I / 57**

## **Plenary 2**

**Morning Session I / 58**

## **Plenary 3**

**Afternoon Session II / 59**

## **Talk 1**

**Afternoon Session II / 60**

## **Talk 2**

**Afternoon Session II / 61**

## **Discussion**

**Afternoon Session II / 62**

## **Talk 3**

**Afternoon Session II / 63**

## **Talk 4**

**Afternoon Session II / 64**

## **Discussion**

**Afternoon Session II / 65**

## **Talk 1**

**Afternoon Session II / 66**

## **Talk 2**

**Afternoon Session II / 67**

## **Discussion**

**Afternoon Session II / 68**



## Talk 3

Afternoon Session II / 69

## Talk 4

Afternoon Session II / 70

## Discussion

Morning Session II / 71

## TDAnalyser - a modular framework for the analysis of test beam data

**Authors:** Berkan Kaynak<sup>1</sup>; Laurent Forthomme<sup>2</sup>; Onur Potok<sup>1</sup>

<sup>1</sup> *Istanbul University (TR)*

<sup>2</sup> *AGH University of Krakow (PL)*

We present a new, modular framework for the processing of test beam data, and in particular for the R&D programme of future timing detectors.

Based on a C++ architecture, it aims to normalise workflows for the analysis of detector performances through the definition of standard and user-defined analyses (e.g. time discrimination algorithms, intrinsic time resolution, or inter-channel correlations extraction).

Its modular extension toolset gives analysts a platform for the definition of their workflow, including the definition of unpacking algorithms for oscilloscopes or DAQ-specific output formats, the combination of multiple sources into a global event content, or the extraction of calibration parameters from a fraction of datasets.

With its RNTuple backbone for the management of the run- and event-granular payloads (including user-defined objects), it provides multiple I/O modules implementations for the preservation and standardisation of test beam datasets.

Thanks to its TGeoManager-based geometry management system, it also allows the visual monitoring of channel occupancy, or the interfacing to external tracking algorithms to produce high-level information from simple waveforms or user-specific collections.

In this talk, a large emphasis is given on the implementation of multiple ROOT-based solutions, including our experience on the usage of RNTuple models for the internal event data model management.

Morning Session I / 72

## Bridging Gravitational Wave and High-Energy Physics Software

**Author:** Marco Meyer-Conde<sup>1</sup>

<sup>1</sup> *Tokyo City University (JP), University Of Illinois (US)*

Gravitational Wave (GW) Physics has entered a new era of Multi-Messenger Astronomy (MMA), characterized by increasing GW event detections from GW observatories at the LIGO-Virgo-KAGRA collaborations. This presentation will introduce the KAGRA experiment, outlining the current workflow from data collection to physics interpretation, and demonstrate the transformative role of machine learning (ML) in some GW data analysis.

This talk also bridge advancements in computational techniques between fundamental research in Astrophysics and High-Energy Physics (HEP). Such initiatives may find some common interests in the context of next generation trigger systems in HEP and advanced signal processing. Innovative solutions for addressing next-generation data analysis challenges will be presented, with a focus on the use of modern ML tools within the ROOT C++ Framework (CERN) and introducing Anaconda HEP-Forge for rapid software deployments. These tools, available as an additional shared libraries in ROOT, integrate key requirements for typical astrophysical analysis, but also HEP physics analysis —such as complex filtering, vector manipulation, KAFKA & other Cloud data transfers, and complex tensor computations on both CPU and GPU technologies.

## Morning Session I / 73

### The outside view on ROOT

**Author:** FERNANDO HUESO GONZALEZ<sup>None</sup>

As a physicist devoted to medical physics research, I've never fitted a Higgs search plot nor run worldwide distributed analysis of data taken at CERN. Unexpectedly, I am heavy user of ROOT, however for more mundane applications related to my research with high count rate radiation detectors, as well as for teaching, at a small laboratory within CSIC/University of Valencia.

Thus, I would like to present the view on ROOT from outside the CERN ecosystem, a potentially much larger user base. I will highlight the key strengths of ROOT for this kind of community, the main hurdles discouraging its cheerful adoption by newbie students, and identify the main aspects that I consider a priority to improve.

Naturally, these might not align with the priorities in the HEP software roadmap. However, I consider that a broad adoption of ROOT and finding a balance between inside-CERN versus outsider support pays back: it increases the likelihood of bug detection, feedback and number of contributors volunteering to improve this excellent open-source software.

## Afternoon session I / 74

### Performance Comparison of Lossless Compression Algorithms on CMS Data using ROOT TTree and RNTuple

**Author:** Simone Rossi Tisbeni<sup>1</sup>

**Co-author:** Silvio Donato<sup>2</sup>

<sup>1</sup> *Universita Di Bologna (IT)*

<sup>2</sup> *Universita & INFN Pisa (IT)*

The High-Level Trigger (HLT) of the Compact Muon Solenoid (CMS) processes event data in real time, applying selection criteria to reduce the data rate from hundreds of kHz to around 5 kHz for raw data offline storage. Efficient lossless compression algorithms, such as LZMA and ZSTD, are essential in minimizing these storage requirements while maintaining easy access for subsequent

analysis. Multiple compression techniques are currently employed in the experiment's trigger system. In this study, we benchmark the performance of existing lossless compression algorithms used in HLT for RAW data storage, evaluating their efficiency in terms of compression ratio, processing time, and CPU/memory usage. In addition, we investigate the potential improvements given by the introduction, in the CMSSW software framework, of RNTuples: the next-generation data storage format developed within the ROOT ecosystem. We explore how the new format can enhance data read efficiency and reduce storage footprints compared to the traditional format. With the upcoming Phase-2 upgrade of the CMS experiment, efficient compression strategies will be essential to ensure sustainable data processing and storage capabilities. This work provides insights into the different compression algorithms and how the new RNTuples data format can contribute to addressing the future data challenges of the CMS experiment.

## Morning Session II / 75

### Vibe Plotting: ServiceX, RDataFrame, and a LLM

**Author:** Gordon Watts<sup>1</sup>

<sup>1</sup> *University of Washington (US)*

Can we teach an LLM to plot experimental HEP data? Modern particle physics workflows increasingly depend on a complex software ecosystem that connects large datasets, distributed data delivery, and user-level analysis tools. We demonstrate how a Large Language Model (LLM) can act as a coding assistant that bridges these components. Starting from a high-level user request—such as “plot jet transverse momentum in dataset X”—the LLM generates code that orchestrates ServiceX for columnar data delivery, configures RDataFrame operations, and produces the plot. The system dynamically adapts code generation to experimental conventions and schema variations, reducing the barrier to entry. We present examples of successful end-to-end workflows, discuss failure modes and reliability issues, and outline the challenges of integrating such assistants into production environments.

76

### RNTuple for SND@HL-LHC

**Author:** Filippo Mei<sup>1</sup>

<sup>1</sup> *Universita e INFN, Bologna (IT)*

With the latest ROOT version (6.36), the RNTuple API is stable as well as its on-disk format: data written today will be readable with the future versions of ROOT. RNTuple is the successor of the TTree format, bringing many advantages such as faster read / write, type safety, use of modern smart pointers, suitability for parallelized hardware and asynchronous operations. SND@HL-LHC will be an upgraded version of the current SND@LHC experiment that will operate during the High Luminosity phase of LHC. As for other Run 4 experiments, using RNTuple seems like the most sound option.

The first step for SND@HL-LHC is dropping the usage of TClonesArray within its software in favor of an event data model based on STL containers, optimised for IO rate. Then, tuning the SND@HL-LHC software stack in order to work with RNTuple.

## Morning Session I / 77

## Designing and evolving a (Py)ROOT based software framework for SHiP

**Author:** Oliver Lantwin<sup>1</sup>

<sup>1</sup> *Universitaet Siegen (DE)*

As a low background experiment at the intensity frontier, the SHiP experiment faces many unique computing challenges compared to collider experiments. At the same time, the experiment is small, with very limited person power, and everything is evolving quickly in preparation for the TDRs, including core parts of the software, which need to be replaced while avoiding disruption to future-proof the framework for the coming two decades.

ROOT and especially PyROOT are core parts of our framework and allow us to present a consistent, pythonic and stable interface to the packages used in the framework, abstracting individual components from the user.

This talk will give an overview of the computing challenges, future directions and the current and future use of ROOT within the collaboration.

**Morning Session I / 78**

## The Utilization of ROOT in the BESIII Offline Software System

**Author:** Jiaheng Zou<sup>1</sup>

**Co-authors:** Ziyang Deng<sup>2</sup>; Xiaobin Ji<sup>3</sup>; Qiumei Ma<sup>2</sup>; Sheng-Sen Sun<sup>4</sup>; Weidong Li<sup>2</sup>; Xingtao Huang<sup>5</sup>; Zhengyun You<sup>6</sup>

<sup>1</sup> *IHEP, Beijing*

<sup>2</sup> *IHEP*

<sup>3</sup> *IHEP, CAS*

<sup>4</sup> *Institute of High Energy Physics, Chinese Academy of Sciences*

<sup>5</sup> *SDU*

<sup>6</sup> *Sun Yat-Sen University (CN)*

The BESIII experiment has been operating since 2009, to study physics in the  $\tau$ -charm energy region utilizing the high luminosity BEPCII (Beijing Electron-Positron Collider II) double ring collider. The BESIII Offline Software System (BOSS) is built upon the Gaudi framework, while also leveraging ROOT extensively across its various components.

The BESIII experiment primarily utilizes ROOT for data management and storage. Most data, including simulated (rtraw), fully reconstructed (REC), and slimmed event data (DST), are stored in ROOT format, while RAW data remains in binary. The ROOT Conversion Service (RootCnvSvc), developed following Gaudi's specifications, enables bidirectional conversion between transient data objects (TDS) and persistent ROOT files. Calibration constants are serialized using ROOT and stored as BLOBs in databases, with functionalities for access provided by the BESIII Experiment Management Platform (BEMP). ROOT is also integral to validation workflows, where histogram-based comparisons serve as a critical benchmark for assessing both the functional accuracy and performance metrics of new software releases. Furthermore, the tag-based analysis software improves physics analysis efficiency by reorganizing DST events to boost the basket cache hit rate for non-contiguous data in ROOT. Many other tools, such as event display, also rely on ROOT.

**Morning Session II / 79**

## CMS FlashSim and ROOT RDataFrame for ML-Based Event Simulation

**Author:** Filippo Cattafesta<sup>1</sup>

<sup>1</sup> *Scuola Normale Superiore & INFN Pisa (IT)*

CMS is developing FlashSim, a machine learning–based framework that produces analysis-level (NANOAOD) events directly from generator-level inputs, reducing simulation costs by orders of magnitude. Efficient integration of preprocessing, inference, and output is essential, and ROOT RDataFrame provides the backbone of this workflow.

Certain operations required for FlashSim are not yet part of the native RDataFrame API. These include event batching to optimize GPU utilization, efficient writing of ML-generated events through the RDataFrame interface, and support for oversampling to reuse inputs across multiple ML inference iterations. We implemented these features as custom extensions, but a native ROOT implementation would provide substantially better performance and scalability.

We present FlashSim through a simplified demonstrator that illustrates these operations and motivates discussion with the ROOT community on possible solutions and future directions.

## Afternoon Session II / 80

### Implementation of an $H \rightarrow \tau\tau$ Analysis within a ROOT-based C++ Framework using ATLAS Open Data

**Author:** Morvan Vincent<sup>1</sup>

<sup>1</sup> *Univ. of Valencia and CSIC (ES)*

Integration of an analysis of Higgs boson decaying to a pair of tau leptons into a ROOT-based C++ framework, making use of recent ATLAS Open Data release. The workflow demonstrates how modern analysis strategies can be easily implemented within a modular structure that combines event selection and object reconstruction. Particular emphasis is placed on the use of ROOT for handling large datasets and producing complex histograms. The framework is designed to be accessible to students and researchers, providing an educational platform while remaining scalable for advanced physics studies. Results are illustrated with reconstructed di-tau invariant mass distributions, highlighting the pedagogical value of the ATLAS Open Data resources using ROOT.

## Morning Session I / 81

### Introduction to the JUNO offline data processing software and its event data model

**Author:** Teng Li<sup>1</sup>

<sup>1</sup> *Shandong University, CN*

The Jiangmen Underground Neutrino Observatory (JUNO) is a next-generation neutrino experiment in south China currently in the commissioning stage. JUNO's primary objective is to determine the neutrino mass ordering (NMO) mainly by detecting reactor antineutrinos.

Unlike typical accelerator experiments, the JUNO experiment typically detects physical events (such as the inverse beta decay of reactor antineutrinos) that generate multiple trigger signals correlated in time. This feature has caused extra complexity when designing its offline data processing software, especially when time-correlated analysis plays a vital role analyzing JUNO data.

This contribution will briefly introduce the design of the JUNO offline software (JUNOSW) that implements the offline data processing and MC simulation of JUNO. In particular, we will introduce

JUNOSW's event data model (EDM) based on TObject that comprehensively correlates event objects by implementing a smart pointer using ROOT's TProcessID mechanism.

## Morning Session II / 82

### NDMSPC: Addressing THnSparse Challenges in High-Dimensional Analysis

**Author:** Martin Vala<sup>1</sup>

<sup>1</sup> Pavol Jozef Safarik University (SK)

The NDMSPC project introduces an innovative THnSparse analysis framework for high-dimensional data, directly addressing the challenge of fitting N-dimensional histograms within memory. Our solution organizes the THnSparse structure into sub-chunks, optimizing memory efficiency and enabling scalable processing. Beyond numerical values, the framework allows bin content to be arbitrary objects, greatly expanding data representation capabilities. A key feature is the ability to execute user-defined jobs for each bin, providing powerful and flexible custom analysis. This presentation will highlight the NDMSPC framework's design, benefits, and applications in advanced scientific data analysis.

83

### NDimensional Visualization in 3D Virtual Environments

**Authors:** Daniel Chovanec<sup>1</sup>; Stefan Korecko<sup>1</sup>

<sup>1</sup> Technical University of Košice (SK)

Thanks to the recent technological advances, immersive virtual reality becomes a viable option in various areas, including data visualization. In this talk, we present NDMVR, a software solution for n-dimensional histograms visualization in web-based virtual reality. While NDMVR utilizes JS-ROOT for certain base functionalities, its distinct features are nested histogram support and reactive approach to internal and user communication.

## Afternoon session I / 84

### User Story: Integration of ROOT RNTuple to CMSSW's SoA data structures

**Author:** Markus Holzer<sup>1</sup>

**Co-author:** Felice Pantaleo<sup>1</sup>

<sup>1</sup> CERN

The *Struct of Arrays* (SoA) layout separates structure fields into individual arrays, each holding a single attribute across all elements. Compared to the traditional *Array of Structures* (AoS), SoA improves data locality, vectorisation, cache usage, and memory bandwidth—critical for heterogeneous computing. In CMSSW, SoA is implemented using the *Boost::PP* library to provide a clean, user-friendly interface. This work outlines CMSSW's SoA features, their integration with *ROOT RNTuple*, and highlights recent developments and planned extensions. We also present CMSSW's requirements for the RNTuple format moving forward.

**Morning Session I / 85****Market Surveillance at Scale: A Deployed ROOT Framework for Financial Integrity****Authors:** Joost Pennings<sup>1</sup>; Philippe Debie<sup>1</sup><sup>1</sup> *Wageningen University*

We bring a core technology from high-energy physics to financial market surveillance. The HighLO project uses the ROOT framework to process terabytes of high-speed trading data where conventional tools fail. In this presentation, we'll showcase our deployed platform and demonstrate how it equips regulators with powerful tools to protect market integrity.

**Morning Session I / 86****Beyond HEP: ROOT for Solar Resources****Authors:** Daniel Perez Astudillo<sup>None</sup>; Dunia Bachour<sup>None</sup>

ROOT's largest user base is in High-Energy Physics and consequently, most of its functionalities cater to this field. This, however, does not mean ROOT has limited or no place in other areas; its many powerful capabilities such as data storage in compressed binary files, data analysis, modelling and simulation, fitting, data display and even machine learning can be exploited in most other fields of science and technology. Astrophysics and Medical Physics are known areas where ROOT is employed, and a few finance applications show up occasionally in the ROOT Forum (<https://root-forum.cern.ch/search?q=finance%20order%3Alatest>).

In this poster I will show examples of the use (and needs) of ROOT for over a decade in the field of solar resource assessment, that is, the analysis of solar radiation and other atmospheric phenomena that affect the available solar energy on the surface of the planet, for applications such as solar power production and climate studies.

**Morning Session I / 87****FCCAnalyses: a ROOT-based Framework for End-to-End Physics Analysis at the FCC****Authors:** Jan Eysermans<sup>1</sup>; Juraj Smiesko<sup>2</sup><sup>1</sup> *Massachusetts Inst. of Technology (US)*<sup>2</sup> *CERN*

We present an overview of the FCC analysis framework, designed to streamline user workflows from event processing to final results. Built on top of ROOT's RDataFrame and the EDM4hep data model, FCCAnalyses provides a coherent environment for dataset processing, visualization, plotting, and statistical fitting. We will highlight the full analysis chain from the user perspective, including different execution modes, distributed computing integration (HTCondor, Slurm, Dask Gateway), and technical integrations such as TMVA and machine learning interfaces. The presentation will conclude with a discussion of current challenges and planned improvements.

**Afternoon Session II / 88**

## ROOT.jl, opening ROOT to the Julia programming language

**Author:** Philippe Gras<sup>1</sup>

<sup>1</sup> *Université Paris-Saclay (FR)*

In a single programming language, Julia provides ease of programming (as Python), high running performance (as C++), and exceptional code reusability (with no comparison). These three properties make it the ideal programming language for high energy physics (HEP) data analysis, as demonstrated by several studies and confirmed by a growing interest from the HEP community. The ROOT.jl package provides a Julia API to the ROOT analysis framework. The number of covered ROOT classes is growing with each release. Its latest release covers the full Histogram and Geometry libraries, as well as support of TTree read/write, and graphic display (TCanvas, TBrowser, and products of the Geometry libraries). It supports the Julia interactive help, based on the ROOT reference manual content. The ROOT.jl features will be presented in this talk. How automatic generation of code and help content is used to minimize the maintenance effort in the integration of new ROOT releases will also be shown.

89

## Cross compilation of the ROOT libraries

**Author:** Philippe Gras<sup>1</sup>

<sup>1</sup> *Université Paris-Saclay (FR)*

Cross compilation allows the build of binaries of a software package for different platforms without the need to use different hardware and operating system for each platform being supported. The Julia binary registry, uses it to ease the support of a number of platforms (ARM, x86, PowerPC combined with MacOS, Linux, and Windows). Support for the Julia binary registry has motivated this work on a cross-compilation method for ROOT. The build process of ROOT, involves building the rootcling executable used to generate the I/O class dictionary source code. Extending the ROOT build system to support cross compilation presents two challenges. The first is the need to build binaries for two different platforms, the host where the build is happening and the target. The second is to extend rootcling to allow the generation on one platform of code that must be run on another platform, in other words, cross-code-generation. So far, cross-compilation for a different C library (glibc/musl) and/or different CPU (aarch64/x86\_64) has been achieved. The compilation is done in an Alpine Linux container. We will present how this cross compilation is performed.

**Morning Session II / 90**

## ROOT in the LHCb Software Framework - past, present and future

**Author:** Marco Clemencic<sup>1</sup>

<sup>1</sup> *CERN*

The LHCb Collaboration has been using ROOT in its software framework from the beginning, in more or less efficient way. Here summarize the evolution of our use of ROOT to then glimpse into the possible paths in view of LHC Run4 and LHCb Upgrade 2.

**Morning Session II / 91**



## Usage of EDM4hep datamodel in RDataFrame

**Authors:** Jan Eysermans<sup>1</sup>; Juraj Smiesko<sup>2</sup>

<sup>1</sup> *Massachusetts Inst. of Technology (US)*

<sup>2</sup> *CERN*

The Future Circular Collider analysis framework, FCCAnalyses, heavily relies on the Key4hep provided datamodel: EDM4hep. The datamodel itself is described by a simple YAML file, and all required I/O classes are generated with the help of Podio, datamodel generator and I/O layer. One of the limitations of the Podio-based datamodel is that its internal structure, while providing convenient containerization of physics objects and resolution of relationships between them, has performance implications when running in the columnar parallelized environment of RDataFrame. The presentation will discuss the approaches undertaken when interfacing ROOT's RDataFrame with the EDM4hep datamodel.

### Morning Session II / 92

## Benchmarking ROOT-Based Analysis Workflows for HL-LHC: The CMS INFN AF Use Case

**Authors:** Luca Pacioselli<sup>1</sup>; Diego Ciangottini<sup>1</sup>; Massimo Sgaravatto<sup>2</sup>; Daniele Spiga<sup>3</sup>; Tommaso Tedeschi<sup>3</sup>

<sup>1</sup> *INFN, Perugia (IT)*

<sup>2</sup> *Universita e INFN, Padova (IT)*

<sup>3</sup> *Universita e INFN, Perugia (IT)*

For a few years, INFN has been developing R&D projects in the context of CMS high-rate analyses in view of the challenges of the HL-LHC phase, starting in 2030. Since then, several studies have been implemented and as a result a prototype for an analysis facility has been proposed, in compliance with CMS analysis frameworks (e.g. based on Coffea, ROOT RDataFrame), and also adopting open-source industry standards for flexibility and ease of use.

All this gave the opportunity to implement systematic studies on ROOT-based workflows at scale. In this context, we present a concrete example of an analysis exploiting ROOT RDataFrame.

Recently, new opportunities in Italy have opened up the possibility of enhancing the benchmarking approach, towards specialised resources (named HPC bubbles) hosted at INFN Padova, featuring three types of heterogeneous nodes: CPU (192 cores, 1.5TB RAM, NVMe) and GPU (4 × NVIDIA H100 80GB). Preliminary results and future plans for detailed studies on metrics such as event rate, I/O operations, energy efficiency, as well as performance comparisons between TTree and RNTuple, and finally validation with a real CMS analysis will be given.

### Afternoon session I / 93

## Highlights of xRooFit - The High-Level API for RooFit

**Author:** Will Buttinger<sup>1</sup>

<sup>1</sup> *Science and Technology Facilities Council STFC (GB)*

xRooFit, created in 2020 and integrated into ROOT as an experimental feature in 2022, is an API and toolkit designed to augment RooFit's existing functionalities. Designed to work with any RooFit workspace, xRooFit adds features to assist with workspace creation, exploration, visualization, and

modification. It also includes a suite of functionality for statistical analysis, including NLL construction and minimization management, dataset generation, profile-likelihood test statistic evaluation, and hypothesis testing (including automated CLs limits with both toys and asymptotic formulae). xRooFit is designed to work with any workspace, no matter how it is created, and works in both C++ and python ecosystems. The primary objective of the xRooFit project is to help users build better, smarter, easier to understand statistical models, and to perform statistical analysis tasks more efficiently.

I will present an introduction to xRooFit, highlighting some of its many functionalities, and showcase the ways that xRooFit has already been seamlessly integrated into many ATLAS statistical analysis workflows.

94

## Cloud-Based Machine Learning with ROOT for the High-Luminosity LHC Era

**Author:** Gurchet Singh<sup>None</sup>

**Co-author:** A Kumar<sup>1</sup>

<sup>1</sup> *Rayat Bahra University Punjab – 140103, India*

Recent advances in the Scientific Python Ecosystem have opened new possibilities for High-Energy Physics (HEP), especially through its integration with ROOT, the backbone framework for data storage, analysis, and visualization. With tools such as NumPy, pandas, and Jupyter now interoperating seamlessly with ROOT, researchers can build flexible and efficient workflows. At the same time, Machine Learning (ML) is becoming essential in HEP, particularly with the High-Luminosity LHC expected to produce exascale datasets. To address these challenges, the MLaaS4HEP framework provides an experiment-independent, cloud-based solution that enables full ML pipelines—from reading ROOT files to model training and inference—via simple HTTP calls. Deployed on INFN Cloud with features such as OAuth2 authentication, XRootD proxy access, and TensorFlow-as-a-Service, this system demonstrates how scalable and distributed ML services can support the next generation of physics analyses.

**Afternoon session I / 95**

## Efficient Parameter Inference with MoreFit

**Author:** Christoph Michael Langenbruch<sup>1</sup>

<sup>1</sup> *Heidelberg University (DE)*

Parameter inference via unbinned maximum likelihood fits is a central technique in particle physics. The large data samples available at the HL-LHC and advanced statistical methods require highly efficient fitting solutions. I will present one such solution, MoreFit, and discuss in detail the optimization techniques employed to make it as efficient as possible. MoreFit is based on compute graphs that are automatically optimized and compiled just in time. The inherent parallelism of the likelihood can be exploited on a wide range of platforms: GPUs can be utilized through an OpenCL backend, CPUs through a backend based on LLVM and Clang for single- or multithreaded execution, which in addition allows for SIMD vectorization. Finally, I will discuss the resulting performance using some illustrative benchmarks and compare with several other fitting frameworks.

**Afternoon session I / 96**

## High performance analysis in CMS

**Author:** Josh Bendavid<sup>1</sup>

<sup>1</sup> CERN

The unprecedented volume of data and Monte Carlo simulations at the HL-LHC poses increasing challenges for particle physics analyses, demanding computation-efficient analysis workflows and reduced time to insight. We present a review of data and statistical analysis models and tools in CMS, with a particular emphasis on the challenges and solutions associated with the recent W mass measurement. We present a comprehensive analysis framework that leverages RDataFrame, Eigen, Boost Histograms, and the Python scientific ecosystem, with particular emphasis on the interoperability between ROOT and Python tools and output formats (ROOT and HDF5). Our implementation spans from initial event processing to final statistical interpretation, featuring optimizations in C++ and RDataFrame that achieve favorable performance scaling for billions of events. The framework incorporates interfaces to TensorFlow for fast and accurate complex multi-dimensional binned maximum likelihood calculations and robust minimization. We will discuss gaps and deficiencies in standard ROOT-based tools and workflows, how these were addressed with alternative integrated or standalone solutions, and possible directions for future improvement.

**Morning Session II / 97**

## Analyzing HL-LHC heavy-ion collision data with ALICE

**Author:** Victor Gonzalez<sup>1</sup>

<sup>1</sup> Wayne State University (US)

Analyzing HL-LHC heavy-ion collision data with ALICE

Victor Gonzalez, Wayne State University (US),  
on behalf of the ALICE Collaboration

The ALICE detector has been taking data at heavy-ion HL-LHC regime since the start of the LHC Run 3 Pb–Pb campaign in October 2023. Recording Pb–Pb collisions at 50 kHz and pp collisions up to 1 MHz interaction rates without trigger results in processing data in real time at rates up to two orders of magnitude higher than during LHC Run 2. To reach that, ALICE underwent a major upgrade on all detectors to make use of the increased luminosity provided by the LHC. The Inner Tracking System now completely consists of Monolithic Active Pixel Sensors which improves pointing resolution. The Time Projection Chamber has been equipped with GEM-based readout chambers to support the continuous readout at the target interaction rate. New forward trigger detectors were installed to allow the clean identification of interactions. The computer infrastructure and the software framework have been completely redesigned for continuous readout, synchronous reconstruction, asynchronous reconstruction incorporating calibration, and a much more efficient and productive scenario for analyzing the considerable increase in the stored data. To reach this point the evolution of the new software framework required key design choices which now constitute the main characteristics of the ALICE Online-Offline, O<sup>2</sup>, computing framework. In this session, the main structure of O<sup>2</sup> as well as its organized analysis infrastructure, Hyperloop train system, will be presented highlighting the features which support the HL-LHC scenario and the physics analysis scenario which provides effective, efficient, and productive access to the huge amount of collected data to the large community of analyzers that conforms the ALICE Collaboration.

**Morning Session I / 98**

## The View from Deep Underground: DUNE's Perspective on ROOT

**Author:** Andrew Paul Olivier<sup>1</sup>

<sup>1</sup> *Argonne National Laboratory*

The Deep Underground Neutrino Experiment (DUNE) plans to collect physics data for over 10 years, starting in 2029. The full DUNE design consists of four far detectors with multiple 10 kT fiducial mass LArTPCs and a heterogeneous near detector complex with 1300 km between them. This combination of technologies and readout time scales is expected to require large storage volumes with on the order of several GBs stored per readout window in the far detector alone. DUNE currently relies on ROOT for a columnar binary data storage format, geometry modelling, fitting algorithms, and more. Before the start of data taking, DUNE is replacing its art framework with the new Phlex framework, with RNTuple support through the FORM I/O toolkit. Additionally, DUNE analysis tools continue to integrate more use of scientific python where possible. Therefore, the development of ROOT 7 is particularly timely for DUNE. This presentation will survey how ROOT performs in DUNE's current software stack and where the future of ROOT can have a meaningful impact on this experiment.

**Morning Session II / 99**

## FORM: Flexible I/O and Storage for the Deep Underground Neutrino Experiment

**Author:** Andrew Paul Olivier<sup>1</sup>

**Co-authors:** Barnali Chowdhury<sup>1</sup>; Peter Van Gemmeren<sup>2</sup>

<sup>1</sup> *Argonne National Laboratory*

<sup>2</sup> *Argonne National Laboratory (US)*

The Deep Underground Neutrino Experiment (DUNE) will deploy four 10 kt fiducial mass time projection chambers in order to study accelerator neutrinos, supernova neutrinos, beyond the standard model physics, atmospheric neutrinos, and solar neutrinos. Reconstructing data in varying time domains over the nearly 400,000 channels needed to monitor this volume presents the complex challenge of processing this information in flexible data-grain sizes suitable for affordable compute nodes. Phlex, a framework under development, will address this challenge using new I/O and storage infrastructure called the Fine-grained Object Reading/writing Model (FORM). In addition to providing flexible, grained I/O, FORM will support several storage technologies, including ROOT's legacy TTree and its replacement RNTuple. This poster will show first results for RNTuple storage for data of varying grain sizes.

**Morning Session II / 100**

## JSRoot in Web Applications for High-Energy Physics: From Interactive Visualizations to Particle Transport Simulations

**Authors:** Leszek Grzanka<sup>1</sup>; Mateusz Bartłomiej Bis<sup>1</sup>; Konrad Michalik<sup>2</sup>; Łukasz Kwinta<sup>2</sup>; Szymon Ciosek<sup>2</sup>; Marek Swakon<sup>2</sup>

<sup>1</sup> *AGH University of Krakow (PL)*

<sup>2</sup> *AGH University Krakow*

This contribution highlights practical uses of ROOT and JSROOT for data exploration and web-based visualization. We show performance improvements achieved by replacing matplotlib with JSROOT

in the CalibView app for the CMS PPS project. A key feature is the partial reading of ROOT files and handling plot data as JSON objects.

We also present a lightweight approach using static websites served with GitHub Pages. By linking to public ROOT files, these pages can visualize large datasets controlled by URL parameters. We compare this method with WebAssembly HDF5 readers, which lack partial read support.

Finally, we discuss JSROOT as the visualization layer of YAPTIDE, a web framework for particle transport simulations. We share insights on user experience and integration within large React-based applications.

Lessons learned from both user and developer perspectives will be shared, including recent developments in uproot that complement web-based workflows.

## Morning Session I / 101

### The FairRoot framework

**Author:** Florian Uhlig<sup>1</sup>

<sup>1</sup> *GSI - Helmholtzzentrum für Schwerionenforschung GmbH (DE)*

FairRoot is a software framework for detector simulation, reconstruction, and data analysis developed at GSI for the experiments at the upcoming FAIR accelerator complex.

Started as a framework for experiments at the FAIR project, it is meanwhile also used by several experiments outside of GSI.

The framework is based on ROOT and Virtual Monte Carlo (VMC) and allows fast prototyping as well as stable usage.

FairRoot provides basic services which can be adapted and extended by the users. This includes easy to use interfaces for I/O, geometry definition and handling to name only few.

Modular reconstruction and/or analysis of simulated and/or real data can be implemented on the base of extended ROOT TTasks.

The design as well as the possible usage of the framework will be discussed.

102

### The R3BRoot framework

**Author:** Pablo Gonzalez Russel<sup>1</sup>

<sup>1</sup> *University of Santiago de Compostela*

The R3B (Reactions with Relativistic Radioactive Beams) experiment at GSI/FAIR is devoted to exploring the properties of nuclei located far from the valley of stability, with particular emphasis on their structure and reaction dynamics. To address these scientific objectives, a highly versatile reaction setup has been developed. This setup combines high detection efficiency, large geometrical acceptance, and excellent energy and momentum resolution, enabling kinematically complete measurements of a broad range of nuclear reactions induced by high-energy radioactive beams produced in inverse kinematics at relativistic velocities.

To support the design, operation, and analysis of the R3B setup, R3BRoot, a dedicated simulation and analysis framework based on ROOT, has been developed. R3BRoot builds upon the FairSoft and FairROOT frameworks, providing a unified platform for detector simulations that covers the full chain from event generation and digitization to reconstruction, as well as both online and offline analysis. Its modular structure is organized around tasks and parameter containers for calibration, mapping, and the storage of reconstruction variables, enabling users to configure detector responses, calibration constants, and analysis strategies in a flexible and reproducible manner.

A key feature of R3BRoot is the implementation of the Virtual Monte Carlo (VMC) interface, which

allows the seamless use of different transport engines (e.g., GEANT3 and GEANT4) without requiring modifications to detector geometries or user code. This abstraction is particularly valuable for benchmarking simulations with different physics models and for ensuring long-term code sustainability. Furthermore, R3BRoot integrates ROOT I/O, visualization, histogramming, and online analysis tools, ensuring full compatibility with the workflows commonly employed in the nuclear and high-energy physics communities.

In this contribution, we present the design principles and current capabilities of R3BRoot, illustrate its use in detector development and data analysis, and discuss recent improvements and applications within the R3B Collaboration.

**Morning Session I / 104**

## **Physics at the HL-LHC scale**

**Afternoon Session II / 105**

## **An overview of ROOT in Python**

**Afternoon Session II / 106**

## **The ROOT Python bindings features**

**Afternoon Session II / 107**

## **Compiler Research (provisional title)**

**Afternoon Session II / 108**

## **Discussion**

**Morning Session I / 109**

## **Innovative Services for Federated Infrastructures - Deployment and Execution in the Computing Continuum**

**Afternoon session I / 110**

## **RooFit overview**

Afternoon session I / 111

### **A look at statistical inference in HEP (Provisional title)**

Afternoon session I / 112

### **CMS Combine (provisional title)**

Afternoon session I / 113

### **ATLAS statistical inference (provisional title)**

Afternoon Session II / 114

### **ROOT software training**

Afternoon session I / 115

### **ROOT I/O Status Overview**

Afternoon session I / 116

### **A first look at RFile**

Morning Session II / 117

### **RNTuple advanced topics**

Morning Session II / 118

### **Future of ROOT I/O**

**Afternoon Session II / 119**

## **The software stack for generic dynamic Python bindings**

**Afternoon Session II / 120**

## **Hands-on Pythonisations tutorial**

**Morning Session II / 121**

## **Workshop discussions and closing**

**122**

## **Workshop registration**

**Afternoon session I / 123**

## **Computing in the 1990s**

**Afternoon session I / 124**

## **30 years of ROOT**