

Taking Stock of the Run 3 Experience Towards ROOT 7

Jonas Rembser¹ for the ROOT Project
6-5-2025



- ▶ Status of the Project: main KPIs
- ▶ Highlights of the releases this year: 6.36.00 and 6.38.00
- ▶ The evolutionary path that will bring us to ROOT 7




The ROOT Project, Today



ROOT: An Open International Collaboration

Open-source and Open-development

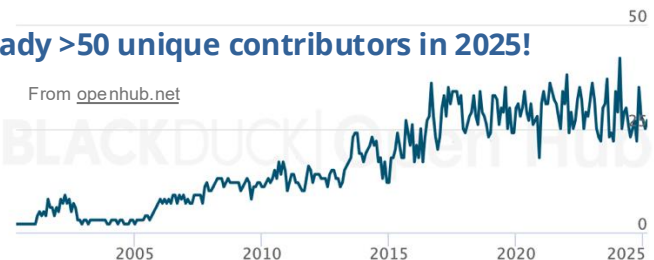
- ▶ [On GitHub](#), LGPL 2.1 
- ▶ PR based model with public review process imposed to core developers and contributors
- ▶ **Very visible authorship of all contributions**

Open-planning: <https://cern.ch/root-pow>

- ▶ **The PoW can be influenced with input, active engagement and contributions!**

**ROOT's open approach pays off:
core team development capacity significantly
amplified by community contributions**

Already >50 unique contributors in 2025!



Per unit time: $N_{\text{contributors}} \approx 1.5 \times N_{\text{Core Dev FTEs}}$
All contribution counts for ROOT!



Short term internships also supported by:





2025 PoW Completion

<https://cern.ch/root-pow>

		Priority	Compl.
	DONE		
	PARTIALLY DONE		
	NOT DONE		
Builds	1 Make all 6.3X releases available on Conda (continuous)	1	0
and	2 Upgrade the Windows builders to Windows 11 & Add MSVC preview builds in the CI	1	0
Binaries	3 Move the ROOT doxy doc generation to the GitHub CI, including its upload for visibility on the web	1	1
	4 Decommission the existing root.cern server in favour of a combination of CERN IT provided services	1	1
	5 Make at least one release available for PIP, bringing this distribution channel to a beta for ROOT	2	0 40 %
I/O	1 Enable schema evolution for std::auto_ptr<T> into std::unique_ptr<T> (2024)	1	1
and	2 Remove the 1GB size limitation for objects written via TTree and row-wise IO (2024)	1	0
TTree	3 Consistency of std::int types across ROOT I/O (needs changes in TTree I/O) (2024)	2	0
	4 Create a new prototype Experimental::RFile (replacement for TFile) that works smoothly with old and new APIs	2	0.25 31 %
RNTuple	1 Take RNTuple classes out of experimental	1	1
	2 Complete the first coherent set of schema evolution features	1	0.75
	3 EP R&D: Implement a demonstrator of arbitrary combinations of chains and friends in the RNTupleProcessor	2	0.75
	4 EP R&D: Design a first version of RNTuple metadata	2	0 63 %
RootFit	1 Numeric integration in n-dim with CUDA (2024)	1	0
	2 EP R&D: Evaluation of custom user functions in CUDA (2024)	1	0.25
	3 Perform analytical minimization of nuisance parameters related to MC statistical uncertainties (upstream from CMS Combine)	1	0
	4 Enable discrete profile likelihood (upstream from CMS Combine)	1	0
	5 Speedup the computation of the Hessian for big Higgs combinations at least by factor of 2 (currently takes several hours)	2	1 25 %
Analysis	1 Reach feature parity of TTree and RNTuple processing with RDF	1	0.25
	2 RDataFrame: enable processing through internal bulk APIs (2024)	1	0
	3 EP R&D: Deliver RDataFrame varied snapshots (2024)	1	0.1
	4 Provide a mechanism to expose objectified NanoAOD preserving lazy reads (2024)	2	0 9 %
Math	1 Improve histos and graphs interoperability with NumPy and UHI protocol and write code examples (2024)	1	1
	2 BMBF: Advance current new histo implementation to one testable by experiments, e.g. integrated at a prototype level with rdf (2024)	1	0
	3 Make numerical algorithms interfaces better accessible from Python e.g. minimisers like Minuit (2024)	1	0
	4 Update the documentation for the Minuit algorithm	2	0
	5 SYCLOPS: Release a library for Lorentz vector computations on accelerators in SYCL (2024)	2	0 20 %
Graphics	1 TScatter2D: Extension of TScatter (4-dim visualisation) to 5-dim	1	0
and	2 Implement auto generated GUI for selected REveElement members	2	0
Visualisation	3 Reduce the time needed and improve user experience of batch image production with web graphics	2	1
	4 Re-write the documentation of the TWebCanvas and RWebWindow classes	2	0.5
	6 REve client's window manager: the undock action to spawn a new window, and increase performance of Geo Browser	2	0 30 %
Interpreters	1 EP R&D Use CppInterOp to replace internals of TClingCallFunc, most notably the need to JIT strings	1	0
	2 SYCLOPS Expose SYCL (prototype) support of ROOT's interpreter	2	0
	4 Review and cleanup whenever possible downstream Clang patches and C++ language extensions	2	0.25 8 %
Substantial extra items not initially foreseen	1 Make RNTupleMerger more robust and make it fully support incremental merging (ATLAS)	1	1
	2 Full support of std::array, e.g. std::SomeContainer<std::array> - ALICE	1	0
	3 Handling of std::conditional by Cling in ROOT (see #16119) - ALICE	2	0
	4 Broken streaming of vector of enum with underlying type other than int (see #16312) - ALICE	1	1
	5		50 %

overall: 31% i.e. 12 / 39 items

► Two priority levels: 1 (high) and 2 (medium)

• Assign a "completion score": 0 not done, 0.25/0.5/0.75 partially done, 1 done.

• Work needed to complete each item varies: assume, that overall it averages out.

► 39 items, unfinished items carried over from 2024: **31% complete**

► 2024 a large PoW: 75 items (including extra items), 65%

► Presented on [January 22](#) and in other slots to allow participation from all timezones.

► **Open-planning: you can influence the content of the ROOT PoW!**



New items added via the engagement of experiments (thanks!)





- ▶ Also in 2025 ROOT continues to deliver what is requested
- ▶ Backlog reduction is *implicitly part of the PoW*
 - Hard to reduce backlog by 40% also during 2025, but so far encouraging results were achieved!

**High development capacity,
keeping up with incoming requests**



- ▶ **Architects Forum (par of WLCG): 10/year**
- ▶ **Quarterly Reports** with experiments and stakeholders
 - Progress and important information broadcasted
 - **Decisions taken.** E.g. release schedule, release contents, actions to take to adapt to changes in the data taking plans
 - Achievement in 2024: obtain updated prioritised list of work items / requests for ROOT every quarter by each experiment
- ▶ **Topical ROOT Meetings (~weekly)**
 - I/O, Librarians and Integrators Meeting (LIM), Statistical Analysis Software, Parallelism Performance and Programming model (PPP)
- ▶ Plus **in person communication**, ROOT's **Mattermost** chat channels, strong presence at **conferences, trainings, hackathons** and, of course, the **ROOT Forum**.

Call for action

Do you represent the software development and integration efforts of your experiment (core, analysis...)?
Do you want to influence ROOT's development with your opinion and needs?

Reach out to us and join the [Quarterly Reports meetings!](#)



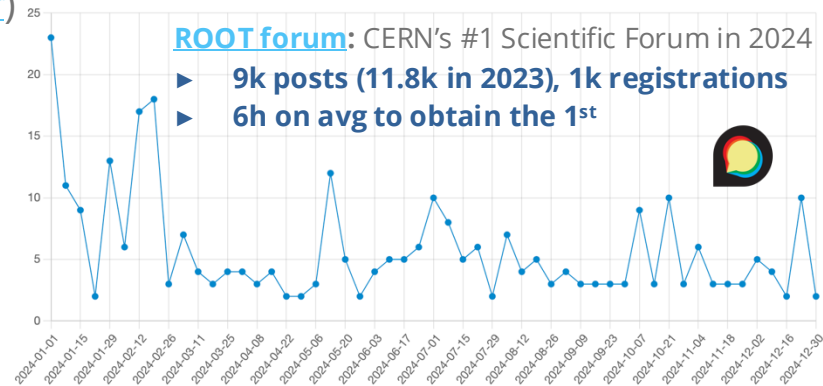
ROOT in the Community: Support, Conferences and Hackathons

2024: ROOT was very present on the international stage, thanks to the support of our FAs

- ▶ 3 at ACAT ([1](#), [2](#), [3](#)), [1](#) at ICHEP, 9 (2 plenaries) at CHEP (all linked [here](#))
- ▶ Workshops, e.g. **WLCG/HSF** ([plenary](#), [parallel](#)), **PyHEP-dev**, and hackathons...
- ▶ ... Plus several contributions about ROOT, not authored by core developers
- ▶ **Two very successful Hackathons** ([February](#), [November](#))



Hackathons: Significantly advance the PoW and connect with our community



ROOT Core devs do not only teach ROOT! E.g. core organiser of the *HEP C++ Course* and *Hands-on Training* – both "Essentials" and "Advanced"

More information about the Hackathons can be found on the [1st](#) and [4th](#) quarterly reports



We scaled up ROOT training: ~300 participants


- ▶ CERN Summer Student Course ~200 participants, 5 instances
- ▶ With IRIS-HEP & HSF: *Python for analysis course* ~90 participants, 3 instances
- ▶ **Platform for teaching: Jupyter Python Notebooks**
- ▶ **We took the opportunity to renovate documentation and training material (e.g. [new video course on CDS](#))**

A new ROOT Video Course is available [on CDS](#): check it out!

ROOT RDataFrame

[RDataFrame documentation](#)

- RDF is ROOT's high-level analysis interface.
- Users define their analysis as a sequence of operations to be performed on the data-frame object;
 - the framework takes care of the management of the loop over entries as well as low-level details such as I/O and parallelisation.
- RDataFrame provides methods to perform most common operations required by ROOT analyses:





Highlights*

*Does not cover RooFit and Analysis, discussed at this workshop [here](#) and [here](#)!



Proposal for 2025: timeline analogous to 2024

▶ **ROOT 6.36.00 Production Release in May 2025**

- [RC1](#) April 23rd
- Data taking release, long support cycle

▶ **ROOT 6.38.00 Development Release in Nov 2025**

- Target analysis and exposing new features
 - Include new histograms and file (in the Experimental namespace)
- ▶ Will agree on a good release schedule for LS3 with experiments in the next months
- Discussions at the quarterly reports will be key



- ▶ **A solid plan, executed and on schedule:**
 - From not being able to store any CMS data products except nano to being able to store all of them
 - Released the 1st stable on-disk format ([CERN-OPEN-2025-001](#)) on time and on budget ([6.34.00 ROOT release](#))
 - RNTuple interfaced extracted from the Experimental namespace according to plans for the 6.36 release
- ▶ We are entering a new phase in the RNTuple life cycle
 - **Transition to production comes with maintenance responsibility: support, training, bug fixes**
 - Slower pace regarding new functionality is expected
- ▶ **We solved enough problems to have confidence that we can solve the remaining ones, too**
- ▶ A successful [RNTuple Workshop took place on December 2-3](#)

The ROOT project thanks experiments for their trust, continued support and collaboration!

In collaboration with experiments, ROOT navigated 6 years of bleeding edge R&D into production!



- ▶ [HEP-CCE Center for Computational Excellence](#) (DOE initiative) carried out a review of RNTuple API
 - Mid-term report delivered in September (can be found [here](#))
- ▶ A rich set of **insightful findings and useful recommendations**
- ▶ **ROOT took those very seriously**, started immediately to address identified deficiencies and implement highlighted potential improvements
 - Many direct benefits of that work will be available in ROOT 6.36.00

HEP-CCE Review: Outlook

HEP-CCE

- RNTuple API Review was a very useful exercise for CCE and the experiments
 - Allowed us to give early feedback on the API we expect to use for years to come
 - And see the feedback being taken seriously. Thanks to the ROOT team!
 - Thanks also to all CCE and non-CCE experts for their work.
- Hopefully beneficial for ROOT experts.
- Looking forward to continued cooperation.



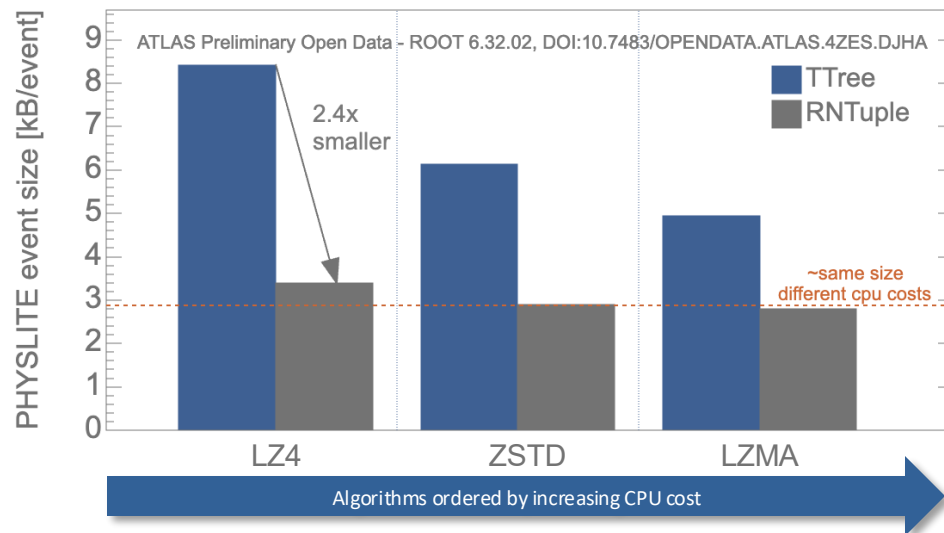
P. Van Gemmeren

RNTuple: an externally reviewed interface!



**RNTuple: smaller files
and more sustainable**

On disk sizes more than 2x smaller than before, saving CPU cycles spent for compression thanks to data pre-conditioning



Nota Bene: size per event not always reduced by factors 2-3. 5% to 20% reductions have been also measured. We believe that such reductions are also excellent results, especially in these early stages of "production RNTuple"!



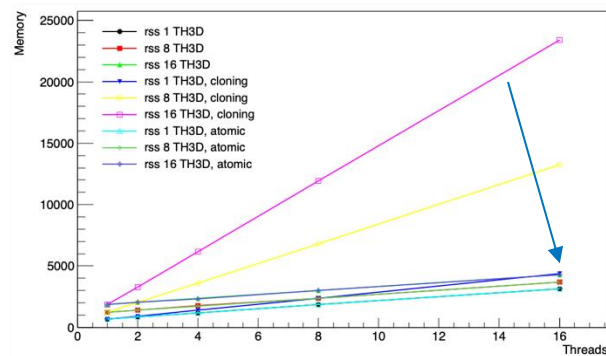
New Histograms for 6.38

- ▶ ROOT 7 will feature a thread-safe and fast histogram class, with sound interfaces
 - Easier to use than the [old RHist prototype](#), and light on templates
- ▶ **Steps to get there in 2025 (Experimental namespace only!):**
 - Remove current RHist prototype in ROOT v6.36: Prototypes are done to learn and improve!
 - Expose new histos in v6.38, also integrated in RDataFrame for fast MT filling
 - Provide known THx functionality by making new histos easily convertible to THx versions

Preparing now for future histograms

Improving existing histograms in 6.38 (CMS requirement)

- ▶ Default RDataFrame behaviour: actions create 1 object per thread (e.g. `myDf.Histo1D`)
- ▶ 3D histos will be shareable among threads **saves memory**
- ▶ Internally thread-safe – preparing for new histos
`ROOT::RDF::Experimental::ThreadsPerTH3(8)`
- ▶ Approach new histograms in steps





Universal Histogramming Interface (UHI)

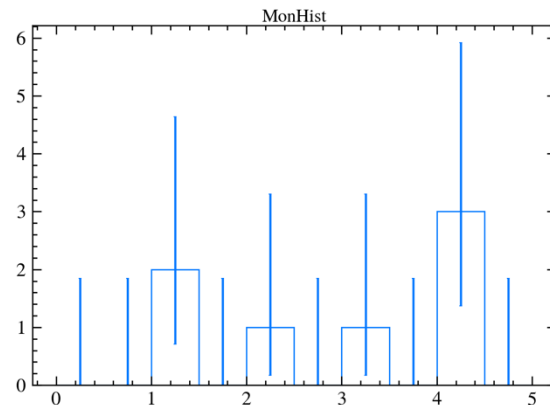
- ▶ [mplhep](#): python tool providing a set of helpers for matplotlib to more easily produce plots typically needed in HEP
- ▶ **UHI protocol:**
 - **Draw ROOT histograms with mplhep** (honouring the UHI protocol)
 - **Indexing for histograms manipulation**
- ▶ **Both drawing and indexing available in ROOT 6.36.00**
 - Thanks to Henry Schreiner for engaging with us about the less clear parts of the specification and for providing a testing suite for it!

```
import mplhep as hep
hep.style.use("LHCb2")
hep.histplot(h)
plt.title("MonHist")
```

a ROOT histogram

```
''' Access'''
v = h[b]
v = h[ROOT.loc(b)]
v = h[ROOT.loc(b) + 1]
v = h[ROOT.underflow]
```

```
''' Setting'''
h[b] = v
h[ROOT.loc(b)] = v
h[ROOT.underflow] = v
h[...] = array(...)
```

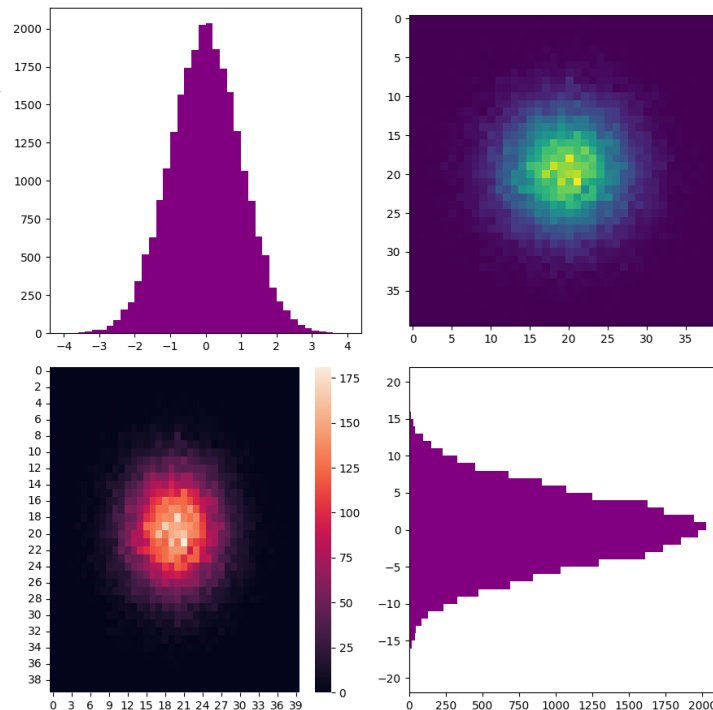




Yet another example of
Python tools and ROOT
interoperability!

```
1 import ROOT
2 from ROOT.ghi import sum
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import mplhep as hep
6 import seaborn as sns
7
8 h2 = ROOT.TH2F("h2", "", 40, -4, 4, 40, -20, 20)
9
10 for i in range(25000):
11     px, py = np.random.normal(0, 1, 2)
12     h2.Fill(px, 5 * py)
13
14 projh2X = h2[:, :, sum, :]
15 projh2Y = h2[:, :, :, sum]
16
17 fig, axes = plt.subplots(2, 2, figsize=(9, 9))
18
19 axes[0, 1].imshow(h2.values(), interpolation='nearest')
20
21 sns.heatmap(h2.values(), ax=axes[1, 0])
22
23 hep.histplot(projh2X, ax=axes[0, 0], histtype='fill', color="purple")
24
25 hep.histplot(projh2Y, ax=axes[1, 1], histtype='fill', color="purple", orientation="horizontal")
26
27 plt.tight_layout()
28 plt.show()
```

Coming next: we are
experimenting with new
approaches, input is welcome!



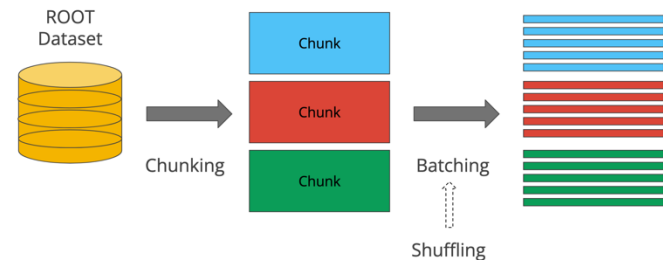


- ▶ TFile served us well: Exabytes of data stored in ROOT files written via the TFile class
 - A rich interface, that grew during the years, also to encompass new technologies (e.g. S3, Davix, XRootD...)
- ▶ Provide a new interface to ROOT files: RFile
 - Previous `Experimental::RFile` prototype removed from the repository for 6.36
 - Prototypes are done to learn and improve!
- ▶ Some principles currently followed:
 - Design with Python usage in mind
 - Reduced interface, easier to comprehend
 - Simple get and put operations
 - Open files written with TFile
 - Clear ownership: no implicit memory management
- ▶ Currently testing whether a directory- or key- based interface is more ergonomic
- ▶ Target v6.38 in November

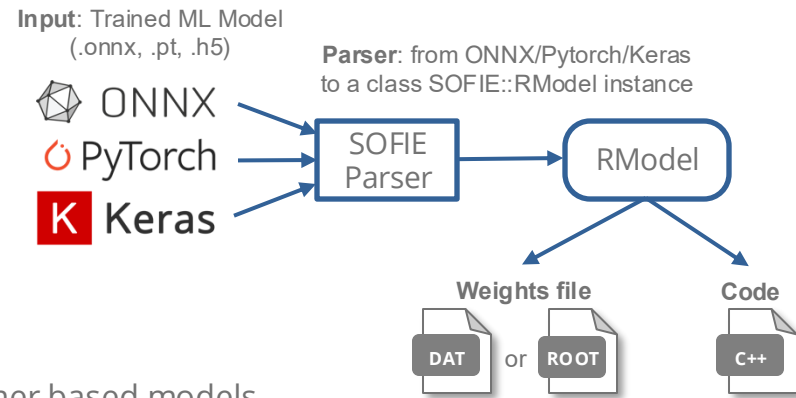


ROOT's ML Strategy: interoperate with existing tools, do not re-invent them

- ▶ Training: Batch generator
 - **Read TTree/RNTuple columnar datasets with RDataFrame**
 - Manipulate data if needed (e.g. filter, define new columns)
 - Feed the model training (e.g. Torch, TF)
 - **No intermediate format, no double-bookkeeping**



- ▶ Inference with SOFIE in C++
 - A **stable, performant and easy** way to run inference in C++
 - **Already available** in ROOT releases
 - Input: ONNX, TF, PyTorch models → output: C++ code & weights
 - Code can be JITted or compiled and distributed as a library
 - Documented rich set of [operators supported](#), GNN and transformer based models
 - **ROOT is ready to add operators needed by experiments**



See [1](#) (training) and [2](#) (inference) at CHEP



- ▶ During the last few months, the core devs and several contributors focussed on **re-organising and improving the set of ROOT code examples (the *Tutorials*)**: entry point [here](#)
- ▶ Almost all **tutorials have a C++ and a Python version** by now
- ▶ Rediscover all examples written during the years: update, modernise, remove, simplify
- ▶ Easier to approach ROOT, learn and teach it by example

Topics | Files

Tutorials

A collection of C++ macros, Python scripts and notebooks helping to learn ROOT by example.

Get started

If you have never used ROOT before and don't know where to start, we recommend that you first explore the [ROOT introductory course](#). You can also watch the recording of the course, but you should follow the material along on your PC. You also may want to have a look at the documentation of these modules:

- [Histograms tutorials](#)
- [Graphs tutorials](#)
- [Fit Tutorials](#)
- [TTree tutorials](#)
- [RNTuple tutorials](#)
- [RDataFrame analysis tutorials](#)
- [RooFit packages tutorials](#)

The \$ROOTSYS/tutorial/s directory includes several sub-directories:

Multidimensional Models

Tutorial		Description
rf301_composition.C	rf301_composition.py	Multi-dimensional pdfs through composition, e.g. substituting a pdf parameter with a function that depends on other observables.
rf302_utilfuncs.C	rf302_utilfuncs.py	Utility functions classes available for use in tailoring of composite (multidimensional) pdfs.
rf303_conditional.C	rf303_conditional.py	Use of tailored pdf as conditional pdfs.s.
rf304_uncorrprod.C	rf304_uncorrprod.py	Simple uncorrelated multi-dimensional pdfs.
rf305_condcorrprod.C	rf305_condcorrprod.py	Multi-dimensional pdfs with conditional pdfs in product.
rf306_condpereventerrors.C	rf306_condpereventerrors.py	Conditional pdf with per-event errors.
rf307_fullpereventerrors.C	rf307_fullpereventerrors.py	Full pdf with per-event errors.
rf308_normintegration2d.C	rf308_normintegration2d.py	Normalization and integration of pdfs, construction of cumulative distribution functions from pdfs in two dimensions.
rf309_ndimplot.C	rf309_ndimplot.py	Making 2/3 dimensional plots of pdfs and datasets.
rf310_sliceplot.C	rf310_sliceplot.py	Projecting pdf and data slices in discrete observables.
rf311_rangeplot.C	rf311_rangeplot.py	Projecting pdf and data ranges in continuous observables.
rf312_multirangefit.C	rf312_multirangefit.py	Performing fits in multiple (disjoint) ranges in one or more dimensions.
rf313_paramranges.C	rf313_paramranges.py	Working with parametrized ranges to define non-rectangular regions for fitting and integration.
rf314_paramfitrange.C	rf314_paramfitrange.py	Working with parametrized ranges in a fit. This an example of a fit with an acceptance that changes per-event.
rf315_projectpdf.C	rf315_projectpdf.py	Marginialization of multi-dimensional pdfs through integration.
rf316_llratiplot.C	rf316_llratiplot.py	Using the likelihood ratio technique to construct a signal enhanced one-dimensional projection of a multi-dimensional pdf.



- ▶ The ROOT core team **is here to support you, listen to your needs and make your data processing and analysis a success!**
 - All sustained by a **long-term support model**
- ▶ Also this year, **ROOT is steadily advancing towards HL-LHC**: ~200 of PRs integrated per month, >50 unique contributors, 19% PoW delivered, -12% items in the trackers
- ▶ **Evolutionary approach bringing to ROOT7**
 - RNTuple interfaces extracted from Experimental mode according to the plan for 6.36
 - Experimental new histograms and file for 6.38
- ▶ Interoperability with Python is a flagship activity
 - Integration with UHI for histograms and batched training
- ▶ Substantial investments in documentation, training and engagement with users to move the community forward!



ROOT Users Workshop: Valencia, 17-21 Nov 2025

- ▶ **Hosted by Universitat Politècnica de València (UPV)**
- ▶ **A welcoming, positive and inclusive atmosphere**
- ▶ **An opportunity to shape together the future of ROOT!**
- ▶ A venue for ROOT users, experts of scientific computing and the ROOT core team to exchange ideas and learn from each other
- ▶ **Emphasis on early-career scientists** and students' perspectives
- ▶ A rich program of presentations, tutorials, and most importantly, discussions. Topics:
 - ROOT roadmap into LHC LS3 and Run 4
 - High performance interactive analysis
 - The new columnar format, RNTuple
 - Evolution of the Python Interface and scientific Python ecosystem
 - Statistical analysis software
 - User feedback

Register today!



ROOT Users Workshop 2025
17-21 Nov 2025

Overview

The 13th ROOT Users Workshop will be held in Valencia, Spain. It is an occasion to discuss ROOT and its related activities today and to help shape the future of the project. The workshop features four and a half days of presentations, discussions, interactions, tutorials, and everything related to ROOT and its various interactions with other HEP software projects. As the end of Run 3 approaches, this workshop is also an opportunity to reflect on how the ROOT project can help address the future computing challenges that HL-LHC and other scientific experiments will provide. In particular, there will be four main topical areas of interest: Analysis, I/O & Storage, Math and Stats, and Scientific Python Ecosystem. The event is foreseen to be in-person only to promote social interactions and the exchange of ideas. To celebrate the community spirit, the event will be complemented by a social dinner and a guided tour of the city of Valencia.

The participation of students is highly valued and encouraged. Hence, a limited number of reduced fee registrations are available for this category. Make sure to register soon to secure your spot!

Event partners

Escuela Técnica Superior de Ingeniería Informática **etsinf**

Starts 17 Nov 2025, 09:00
Ends 21 Nov 2025, 12:00
Europe/Zurich

Organisers
Pedro Alonso Jordá
Marta Czurylo
Vicente Eduardo Padulano
Daniel Pigaro

There are no materials yet.

Application
Application for this event is currently open. [Apply now >](#)

[Help](#) | [Contact](#) | [Terms and conditions](#) | [URL Shortener](#) | [Privacy](#)

<https://cern.ch/root2025>



Backup

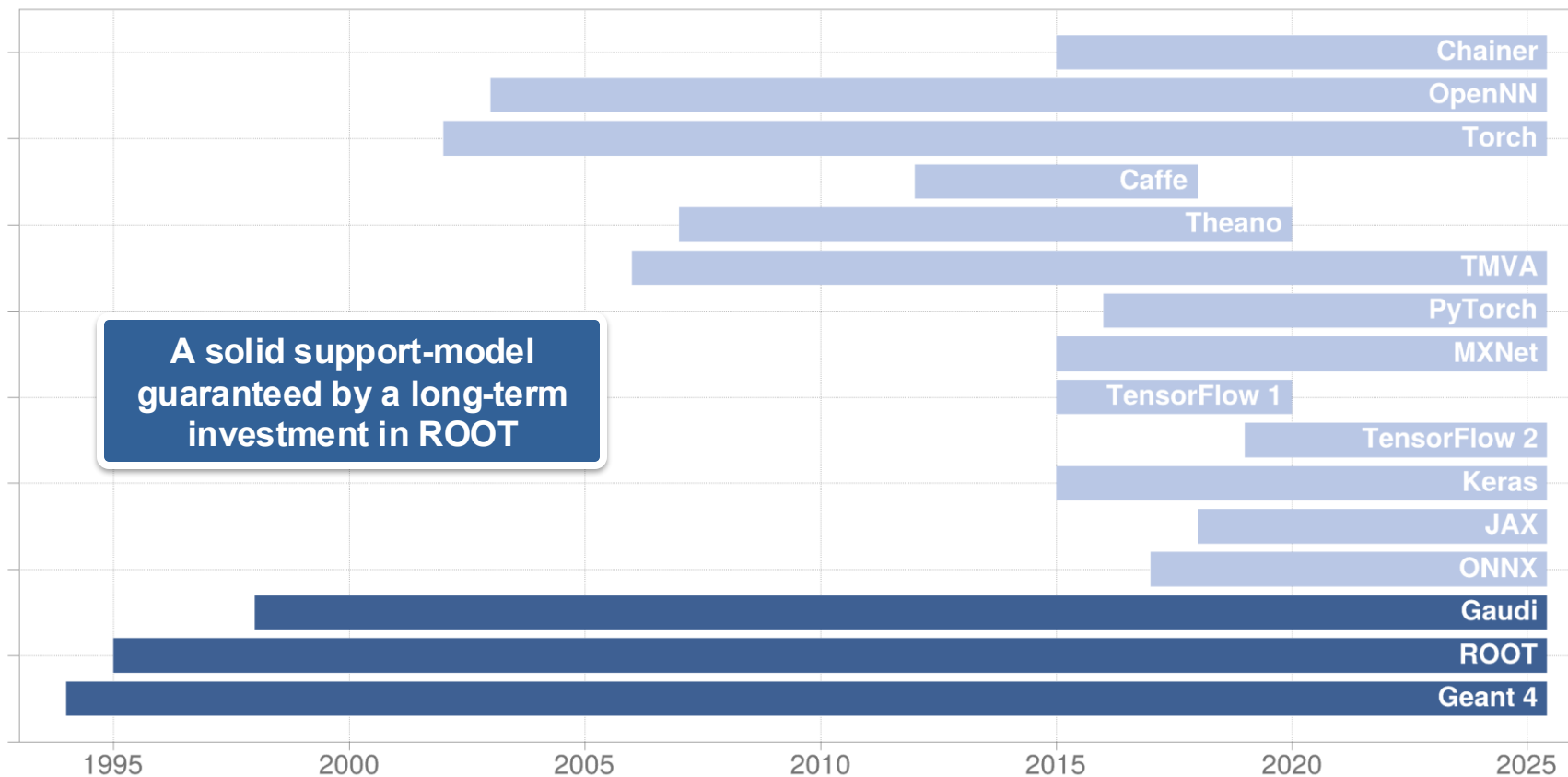


Provide a unified software package for the storage, processing, visualisation and analysis of scientific data that is reliable, performant, supported and sustainable, that is easy to use and obtain, and that minimises computing resources and scientists' time needed to achieve results.

The success of experiments and all ROOT users at large is our priority



HEP Common Software Support Timeline





- ▶ **ROOT is its user community, contributors, and developers**
- ▶ **ROOT is an international collaboration**
 - Institutional responsibilities, but also precious contributions coming from the user community!

Core Devs from <https://root.cern> , PhD students onwards



From left to right, starting from the first row (the affiliation is CERN if not specified):

1. Devajit Valaparambil, Bertrand Bellenot, Danilo Piparo, Florine de Geus, Jakob Blomer
2. Jonas Hanfeld, Jonas Rembser, Marta Czurylo, Olivier Couet, Philippe Canal (**FNAL**),
3. Vasil Vassilev (**Princeton**), Lorenzo Moneta, Monica Dessole, Vincenzo Padulano, Serguei Linev (**GSI**)
4. Giacomo Parolini, Stephan Hageböck, Martin Føll (**Uni Oslo**), Lukas Breitwieser, Silia Taider

Plus students and guests, working with us a few months, up to ~1 year
Not everybody in this slide is 100% dedicated to one project, but many are.



New!