RootInteractive tool for multidimensional statistical analysis, machine learning and analytical model validation

Marian Ivanov[1,*], *Marian* Ivanov[2,**], and *Giulio* Eulisse[2,***]

[1]GSI Darmstadt
[2]UK Bratislava
[3]CERN

Abstract. At the LHC at CERN, ALICE [1] is the experiment optimised for the study of heavy ion collisions. The upgraded ALICE will cope with a tenfold increase in Pb–Pb luminosity and a dramatic increase of two orders of magnitude in the minimum bias events from Pb–Pb collisions in Runs 3 and 4. To cope with this high detector occupancy and event pile-ups, advanced multidimensional data analysis techniques and machine learning are essential.

Machine learning (ML) has become very popular in multidimensional data analysis in recent years. Compared to the simple, low-dimensional approaches used in the past, it is more difficult to interpret machine learning models and evaluate their uncertainties. On the other hand, oversimplification and reduction of dimensionality in the analysis often lead to wrong results.

Our goal was to provide a tool for dealing with multidimensional problems, to simplify data analysis in many (optimally all relevant) dimensions, to fit and visualize multidimensional functions including uncertainties and biases of their representation, to validate assumptions and approximations, to easily define the functional composition of analytical parametric and non-parametric functions, to exploit symmetries and to define multidimensional "invariant" functions and corresponding alarms.

RootInteractive [2] is a general-purpose tool for multidimensional statistical analysis. We use a declarative programming paradigm where we build the structure and elements of computer programs and express the logic of a computation without describing its control flow. This approach makes it easy to use for domain experts, students, and educators. RootInteractive provides functions for interactive, easily configurable visualization of unbinned and binned data, interactive n-dimensional histogramming/projection, and derived aggregate information extraction on the server (Python/C++) and client (Javascript). We support client/server applications using Jupyter, or we can create a stand-alone client-side application/dashboard.

Using a combination of lossy and lossless data compression, datasets with, for example, O(10^7) entries times O(25) attributes can be analyzed interactively in the standalone application in the O(0.500-1 GB) browser. By applying a suitable representative downsampling O($10^{-2} - 10^{-3}$) and subsequent reweighting or pre-aggregation on the server or batch farm, the effective statistics corresponding to one year of ALICE data taking can be can be analyzed interactively

*e-mail: marian.ivanov@cern.ch
**e-mail: marian.i@cern.ch
***e-mail: Giulio.Eulisse@cern.ch

in many dimensions for calibration/reconstruction validation, QA/QC or statistical/physical analysis.

1 Introduction

RootInteractive, developed within the ALICE collaboration, is a tool for advanced multidimensional interactive statistical analysis. It effectively addresses the challenges of the high interaction rates of ALICE data taking during LHC Run 3. These challenges include for example event pile-up, space point distortions in the Time Projection Chamber (TPC) detector due to the accumulated space charge, electronic baseline fluctuations in the TPC, and other distortions. A deep understanding of the detector system, MC simulations and calibration performance is essential for effective use of machine learning in physics analysis. Our goal is to provide a solution that simplifies multidimensional data analysis:

- Fitting and visualizing N-dimensional functions taking into account uncertainties and biases.
- Streamline validation of assumptions, numerical evaluations and differential model comparisons, enabling function composition for different mathematical functions and error propagation.
- Rapid feedback, reducing analysis time from weeks to seconds for interactive expert discussions.
- Support for multi-dimensional parametric optimization.
- User-friendly configuration options for visualisation of unbinned and binned data, interactive multidimensional histograms and projections. It also allows the derivation of aggregated information accessible on both server (Python/C) and client (JavaScript) platforms.
- To facilitate the creation of stand-alone client-side applications (HTML documents) without the need to install additional software.

RootInteractive's core philosophy, encapsulated in the motto "seeing is believing", emphasizes the importance of queries, iterative interactions, and differential comparisons in understanding complex data. Within ALICE, RootInteractive plays a central role in expert projects such as digital signal processing for Run 3 [3], optimization and validation of track reconstruction for Run 2 and Run 3 [4], MC/data mapping, TPC data volume studies, and differential quality assurance and quality control. It has also been used in the development of the particle identification algorithms and magnetic monopole reconstruction studies (in collaboration with the DUNE experiment) for high dE/dx (mean energy loss per distance traveled), low momentum, and spallation product tracking. In summary, RootInteractive stands at the forefront of advanced data analysis in the challenging scientific context of ALICE CERN.

2 Exploring Symmetries, Alarms, and Invariants in Multi-Dimensional Data Analysis

In our project, we focus on the use of symmetries and invariants (within uncertainties) for multidimensional data analysis. We optimize the handling of normalized data that includes data-analytical model, MC-real data, data-symmetry, data-reference data, and data-machine learning prediction across different dimensions. Data normalization leads to reduced RMS scatter and the ability to implement alarms and outlier detection based on statistical significance, e.g. identifying cases where (data model) exceeds Nσ or using likelihood-based methods. These methods are actively applied in various ALICE projects and include considerations of temporal invariance (e.g. referencing the data to an average run), spatial invariance (taking into account rotational and mirror symmetry), magnetic field symmetry, comparing the data with analytical models, evaluating different machine learning models and assessing the smoothness of the data.

As an ALICE application example of symmetry in data and ML models, we train models where we assume symmetry and compare them to a regression or statistical aggregation without the symmetry assumption. We do this by including or not including the variable with the expected symmetry in the Macchine learning regression or aggregation.

For example, in the case of space point distortion, the particle production is φ-symmetric, so the space charge density in the TPC detector is also φ-symmetric, and accordingly the E-vectors of the distortion should also have the same symmetry. A deviation from symmetry either indicates a problem, or the symmetry is broken by an additional effect that we have to take into account (e.g. a non-φ-symmetric conversion factor from ionisation to space charge) in correction respectivelly in MC simulation. After correcting (normalising over the functional composition) for the known effects, the symmetry should be restored and data consistency can be assessed.

3 RootInteracive - Machine learning, ML validation and data aggregation

RootInteractive integrates external machine learning models and provides wrappers for models such as RandomForest [5] and Extreme Gradient BDT [6] to estimate local parameters of the PDF function (which are used as estimators for the reducible and irreducible error machine learning model). For example, in optimizing the TPC space point distortion model developed by ALICE, we used RootInteractive to compare an external U-net model [7] with a simpler data-driven approach using RandomForest. This involved optimizing the parameters of the models and the cost functions.

RootInteractive performs interactive visualization, data aggregation and invariance validation on the client side, processing significant amount of unbinned data, typically between $O(10^6)$ and $O(10^8)$ elements (rows x attributes). The upper limit depends on memory and data transfer and is typically O(1GB) on the client side. To access even larger amount of data, two main approaches are used: domain-specific sampling (e.g. in ALICE, it provides more uniform momentum or particle type distribution) and pre-aggregation on the server. This pre-aggregated or skimmed information can later be reweighted in subsequent RootInteractive sessions on the client. Pre-aggregated data sources typically include local statistics such as mean, median, count and standard deviation obtained from unbinned predictions (from Machine learning regression). Alternatively, we aggregate data using local kernel regression parameters tabulated on a regular mesh. In RootInteractive we use C (HistoND) and Python libraries like Pandas [8] and Modin [9] for multidimensional operations like 'group by' and rolling statistics.

4 RootInteractive statistics and Machine learning wrappers

RootInteractive provides simple local statistics on the regular grid, including mean, median, RMS, and quantiles, for model validation on the client side (see example snapshot 1).

We also introduce new features:

- Generalized kernel linear regression on the client, similar to that on the server. This uses multi-dimensional group-by, rolling statistics, and local kernel fitting to represent smooth functions.

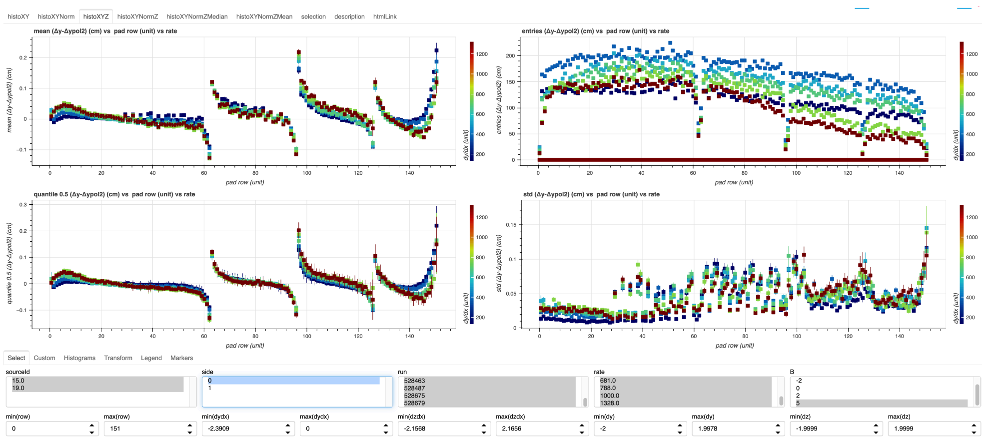

Figure 1. Example of an interactive dashboard created with the getDefaultVars function of the RootInteractive template. The goal of this dashboard is to compare the ALICE TPC space point distortion estimator based on two factors: position in the detector (pad row) and rate (represented by the colour axis). The layout of the dashboard is very versatile. It offers views for 1D, 2D and 3D aggregation, both with and without data normalisation. These different views can be accessed via tabs so that the aggregated data and the different iterations of the machine learning predictions can be easily compared. On the snapshot you can see the 2D layout. It shows important statistical quantities such as mean, median and root mean square (rms) for a given selection (e.g. side, magnetic field, position, etc.).

- Ongoing development of client-side ML prediction using WebAssembly (wasm) [10] and ONNX [11]. This allows us to parameterize machine learning models and define parameterized derived variables. For example, we can compute derivatives of machine learning predictions, perform systematic error studies on the client side, and perform numerical derivatives with varying input parameters of machine learning models.
- A local linear forest, which is a local linear regression with a kernel defined by a random forest, was originally introduced in GRF (Generalised Random Forest) [12]. However, the original implementation is very computationally intensive when it comes to predictions. To mitigate this, we are working on a cached version that incorporates local derivatives in the nodes, similar to the approach used in the previous ALICE software framework [13]) framework in the AliNDLocalRegression class on the fixed grid. This optimisation makes the prediction process more efficient.

5 RootInteractive

The framework presented here, RootInteractive, serves as a versatile tool for interactive statistical aggregation and visualisation of multidimensional data, compatible with both ROOT[1] and native Python data frame formats such as Pandas [8] and Modin [9]. The code provides extensive support for various ROOT data structures and classes, including TTree, TTreeFormula, Aliases, TFormula and static Root/AliRoot functions. Work is also being done to simplify compatibility with RDataFrame [14, 15] and awkward (PyHep) arrays [16]. One can use this framework with various data sources, including PyRoot (AliRoot/O2) data structures. Importantly, it works seamlessly with pandas/modin alone, so one does not need to install the ROOT package. Internally, these data structures are converted into either Bokeh [17] CDS

(ColumnDataSource) for simple scatter visualization or our own RootInteractive CDS extension, which allows for a variety of operations with N-dimensional histograms, projections, and aggregated data.

The content of RootInteractive includes the following main aspects:

- It provides an interactive and highly customisable visualisation solution for both non-binned and binned data.
- It allows interactive operations such as n-dimensional histograms, projections and the extraction of derived aggregated information.
- The application can be used in different configurations, e.g. as a client/server application integrated with Jupyter and Bokeh.
- Users can use the Bokeh standalone dashboard independently as a client application, without having to install software or require a stable internet connection to a server. This is the most common way to use RootInteractive. We can include interactive dashboards as a valuable resource in meeting agendas, for example.
- The system supports both lossy and lossless data compression, enabling efficient data transfer from server to client. In typical use cases, a factor 10 is achieved.
- RootInteractive interfaces seamlessly with ROOT - RDataFrame -awkward - Pandas/Modin tools.
- Work is underway to further simplify RDataFrame's C syntax by using Domain-specific language for slicing and joins (inspired by Python syntax and Pandas/Modin joins). The Python-like syntax is translated into the C++ template functions and used as JIT or in C macros.

This code empowers interactive visualization, histogramming, and data aggregation in N-dimensions on the client side, facilitating advanced data analysis tasks.

6 Interactive visualization, histogramming, and data aggregation in N-dimensions on client

Interactive visualisation, histogram generation and N-dimensional data aggregation on the client side are controlled by a series of Python dictionaries and arrays. These declarations serve as inputs to the bokehDrawSA function, which can be used to create a variety of graphical elements such as scatter plots and N-dimensional histograms, as well as projection statistics, whether they are binned or unbinned. Essentially, these declarations define the data sources for bokeh and allow the development of derived variables and aggregated statistics to enrich the client-side visualization. bokehDrawSA uses declarative programming, an approach that allows developers to express computational logic without having to explicitly script every step of the process. This methodology simplifies the programming effort because developers only need to describe the desired program results, rather than specifying in detail each command or step to achieve those results. In practice, the configuration of the interactive visualization relies on six arrays/dictionaries, as shown below:

```
bokehDrawSA.fromArray(df,selection,figureArray,widgetParams,
layout=figureLayoutDesc, tooltips=tooltips,
parameterArray=parameterArray,
widgetLayout=widgetLayoutDesc, sizing_mode="scale_width",
nPointRender=300,aliasArray=aliasArray,
histogramArray=histoArray,arrayCompression=arrayCompression)
```

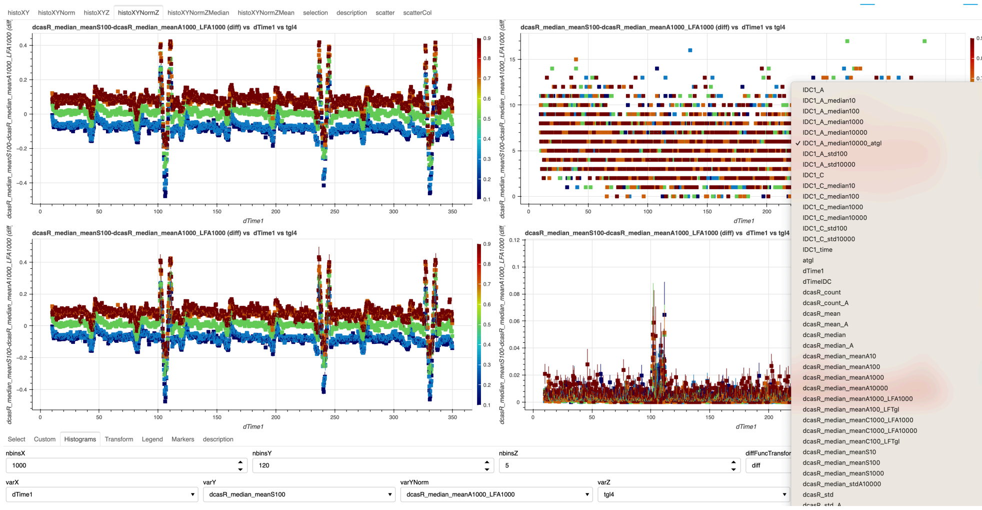

Figure 2. A snapshot of a real use-case 1 interactive dashboard within ALICE, generated using getDefaultVars. This dashboard is used for comparing ALICE TPC [18] currents (IDC) and TPC tracking bias (<DCA>) time series. The layout includes functions for 1D, 2D, and 3D aggregation, facilitating comparisons between aggregated data and various iterations of machine learning predictions. Additionally, it provides local aggregated statistics. You can find a partial list of the input variables for interactive aggregation on the right side of the snapshot (expanding select for the varZ selection)

These arrays, in particular figureArray, histogramArray, aliasArray, layout, widgetLayout, and parameterArray, collectively contribute to the construction and creation of interactive visualizations and provide a streamlined approach to the complex display and analysis of data. The parameter names used in the declaration directories conform to the bokeh naming convention for graphical elements. For histogramming and statistical aggregation, we have taken inspiration from Numpy and Pandas.

To simplify the creation of interactive visualizations, we use "predefined template functions" that provide predefined configurations (dictionaries) that can later be extended in the user's Python code. Our main goal is to simplify the process of automatic multidimensional differential validation, and to this end, we have developed several template functions with parameterisable differences of function, reference function and scale function.

One of these functions, getDefaultVars, is tailored to create default variables suitable for multidimensional data and automatic user-defined normalization. It provides configurations such as aliasArray, variables, parameterArray, widgetParams, widgetLayoutDesc, histoArray, figureArray and figureLayoutDesc.

```
def getDefaultVars(normalization=None, variables=None, defaultVariables={},
                                                weights=None, multiAxis=None)
```

With this function, we create a predefined layout containing 1D, 2D and 3D aggregations (a user extension must create an n-dimensional selection layout). The parameters for aggregation and normalization are based on the lists of input variables. The client-side normalization process provides flexibility and supports different methods such as "delta," "ratio", "log ratio" and pulls (Δ/σ). In example snapshot fig. 2, the diff function is selected using multiselect widget - difFunction.

In example above the following code for aliasArray and parameterArray was generated:

- Generated aliasArray:

```
[{'fields': ['varY', 'varYNorm'],
'name': 'diffFunc',
'parameters': ['diffFuncTransform'],
'v_func':  """
     if($output == null || $output.length !== varY.length){
         $output = new Float64Array(varY.length)
     }
      if(diffFuncTransform=='diff'){
         for(let i=0; i<$output.length; i++){
             $output[i] = varY[i]-varYNorm[i]
         }
     }
      else if(diffFuncTransform=='ratio'){
         for(let i=0; i<$output.length; i++){
             $output[i] = varY[i]/varYNorm[i]
         }
     }
      else if(diffFuncTransform=='logRatio'){
         for(let i=0; i<$output.length; i++){
             $output[i] = Math.log(varY[i])/Math.log(varYNorm[i])
         }
     }
     return $output
  """ }]
```

- Generated parameterArray subset:

```
  {'name': 'diffFuncTransform', 'value': 'diff', 'options': ['diff', 'ratio
                                       ', 'logRatio']},
  {'name': 'varY', 'value': 'dLX_neg_3000_mean_G0', 'options': ['
                                      dLX_neg_3000_mean_G0', '
                                      dLX_neg_3000_mean_G1', ....
```

7 Conclusions

RootInteractive was developed as a tool that simplifies multidimensional data analysis and allows us to effectively process data in all relevant dimensions. This tool aims to fit and visualise multidimensional functions, including uncertainties and biases, validate assumptions and approximations, facilitate the composition of parametric and non-parametric functions, and use symmetries to define multidimensional "invariant" functions and alarms.

RootInteractive offers functions for interactive visualization of both unbinned and binned data, n-dimensional histogramming and projection, and extracting aggregate information, both on the server (Python/C++) and client (JavaScript) in browser. By employing a combination of lossy and lossless data compression techniques, RootInteractive allows interactive analysis of datasets containing millions of entries and dozens of attributes within a web browser, typically consuming around 0.5-1 GB of memory. Through representative downsampling (typically 1-0.1% of data) followed by reweighting or pre-aggregation on the server or batch farm, it enables interactive multidimensional analysis of ALICE's extensive monthly/annual statistics for calibration, reconstruction validation, quality assurance, quality control, and statistical/physical analysis.

Thanks to its versatility and ease of use, RootInteractive plays a central role in ALICE expert projects such as digital signal processing for Run3, optimisation and validation of reconstruction for Run2 and Run3, MC /data mapping, TPC data volume studies and differential quality assurance and quality control. It has also been used in the development of the particle identification algorithm and in magnetic monopole reconstruction studies (in collaboration with the DUNE experiment) for high dE/dx (mean energy loss per distance travelled), low momentum and spallation product tracking.

References

[1] K. Aamodt, A.A. Quintana, R. Achenbach, S. Acounis, D. Adamová, C. Adler, M. Aggarwal, F. Agnese, G.A. Rinella, Z. Ahammed et al., Journal of Instrumentation **3**, S08002 (2008)

[2] M.I.j. Marian Ivanov, *Rootinteractive*, https://github.com/miranov25/RootInteractive (2021)

[3] J. Alme et al. (ALICE TPC) (2023), 2304.03881

[4] M. Arslandok, E. Hellbär, M. Ivanov, R.H. Münzer, J. Wiechula, Particles **5**, 84 (2022)

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., Journal of Machine Learning Research **12**, 2825 (2011)

[6] T. Chen, C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, NY, USA, 2016), KDD '16, pp. 785–794, ISBN 978-1-4503-4232-2, 11

[7] O. Ronneberger, P. Fischer, T. Brox, *U-net: Convolutional networks for biomedical image segmentation* (2015), 1505.04597, 11

[8] W. McKinney, *Data Structures for Statistical Computing in Python*, in *Proceedings of the 9th Python in Science Conference*, edited by S. van der Walt, J. Millman (2010), pp. 51 – 56

[9] D. Petersohn, R. Zadeh, M. Zaharia, X. Meng, E. Smith, J.W. Kottalam, R. Liaw, A. Ghodsi, I. Stoica, *Modin: Scale your pandas workflows by changing one line of code*, https://github.com/modin-project/modin (2019), accessed: 2023-09-04

[10] *WebAssembly Core Specification*, https://www.w3.org/TR/wasm-core-2/

[11] O.R. developers, *Onnx runtime*, https://onnxruntime.ai/ (2021)

[12] S. Athey, J. Tibshirani, S. Wager, The Annals of Statistics **47**, 1148 (2019), 1610.01271

[13] A.S. Framework, *Aliroot*, https://github.com/alisw/AliRoot (2021)

[14] D. Piparo, P. Canal, E. Guiraud, X.V. Pla, G. Ganis, G. Amadio, A. Naumann, E. Tejedor, *Rdataframe: Easy parallel root analysis at 100 threads*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 06029

[15] E. Guiraud, J. Blomer, S. Hageboeck, A. Naumann, V. Padulano, E. Tejedor, S. Wunsch, *RDataFrame enhancements for HEP analyses*, in *Journal of Physics: Conference Series* (IOP Publishing, 2023), Vol. 2438, p. 012116

[16] I. Osborne, J. Pivarski, arXiv preprint arXiv:2302.09860 (2023)

[17] Bokeh Development Team, *Bokeh: Python library for interactive visualization* (2018), https://bokeh.pydata.org/en/latest/

[18] J. Alme et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **622**, 316 (2010)