REST v2.1

A data analysis and simulation framework for micropatterned readout detectors.

RD51 Open Lectures and Mini Week

J. Galan
DPhN/IRFU - CEA-Saclay (France)

11-15 December 2017
CERN
Europe/Zurich timezone

Rare Event Searches for TPCs software

REST arises as a need to unify common codes for simulation and data processing within the group at University of Zaragoza.

REST is motivated by its use in different projects and experiments for rare event searches (as TREX-DM, IAXO-D0, PandaX-III, …).

Collaborative development philosophy, given its high modularity. Major contributions from different institutes : Univ. Zaragoza, SJTU and CEA Saclay.

REST is a development environment based on ROOT for **data analysis and MC simulation**, implementing dedicated classes for MPGD detectors.

REST in a nutshell

REST defines a ROOT-ified and encapsulated metadata and data format.

Fixes basic event data structures : HitsEvent, TimeSignalEvent, TrackEvent

Prototypes the implementation of processes that allow to transform event data.

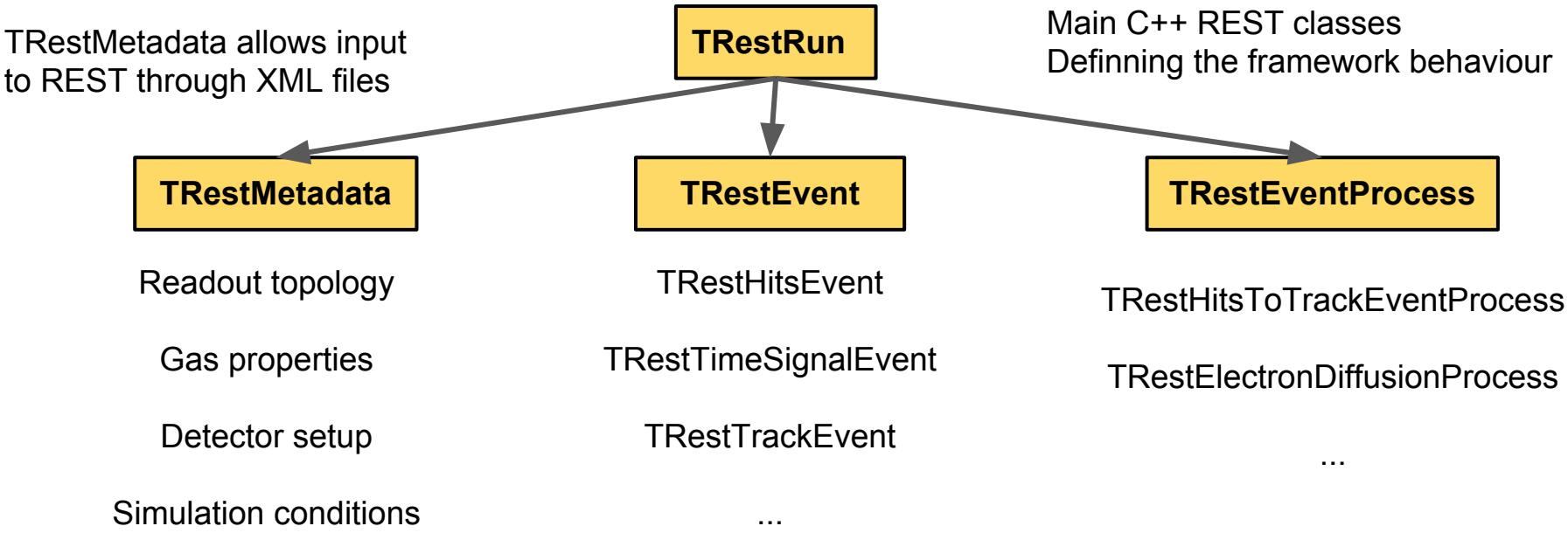
Defines how metadata information is stored and given to the framework to define the behaviour of the framework (allowing to define for example, processes parameters, simulation conditions, detector readout topology, etc).

REST includes pre-defined metadata (readout topology - gas properties) structures and processes that can be used for gas Micropatterned readout detectors.

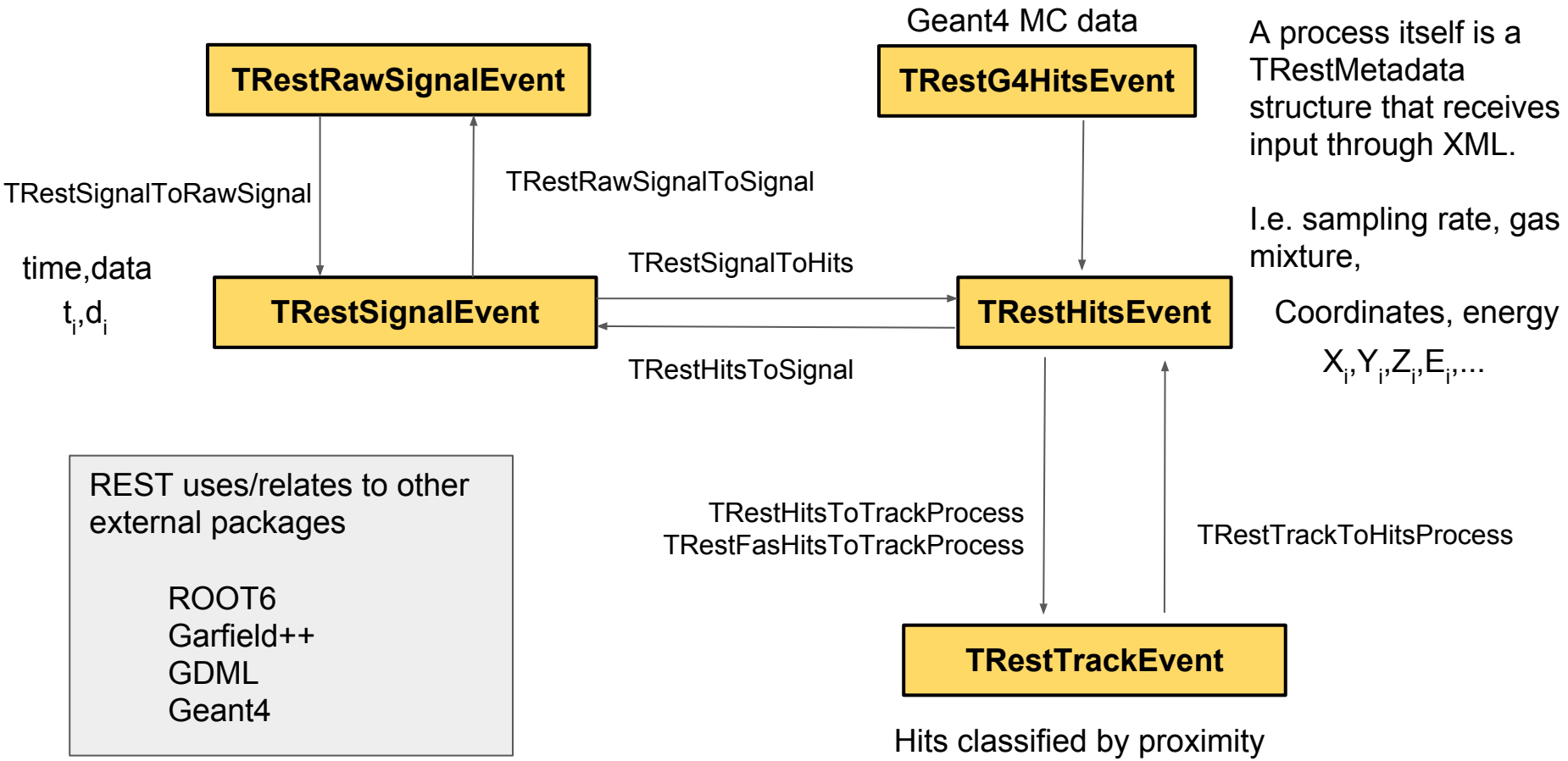
REST Framework scheme

All the event data and metadata access (I/O) is managed/centralized in TRestRun.

TRestRun is also responsible to manage event processing.



Event data type interconnectivity



Framework scalability and structure

REST offers scalability and integration of new metadata, event data types and event processes provided by the user. → User REST library

Common event types, metadata structures and processes related to event reconstruction, detector response and track analysis are already existing inside the framework.

REST framework

Defines basic event types, metadata classes, and basic event data transformation.

+ visualization tools, process manager, common plotting tools, analysis tree, etc

Track processing package

Signal processing package

….

+ user defined

The functionality of the framework is extended using packages that contain related processes.

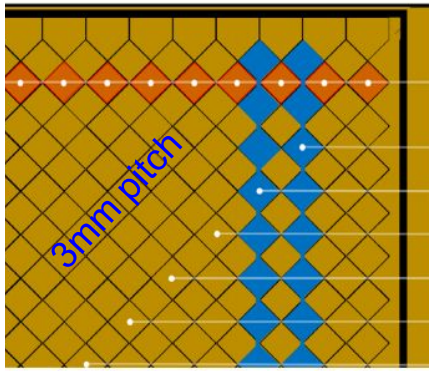
I.e. adding signal noise, signal deconvolution, signal shaping, etc.

TRestReadout metadata definition (hits ↔ signal)

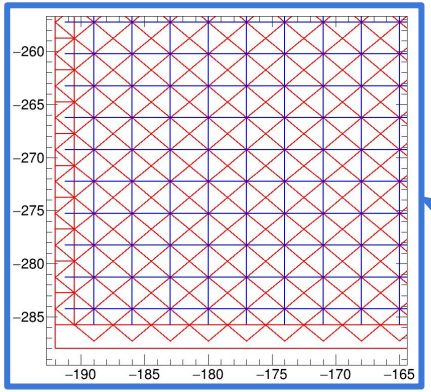
Advanced features in configuration scheme defined by TRestMetadata allows for complex readout topology definitions.

Using FOR loops or complex mathematical expression evaluation.

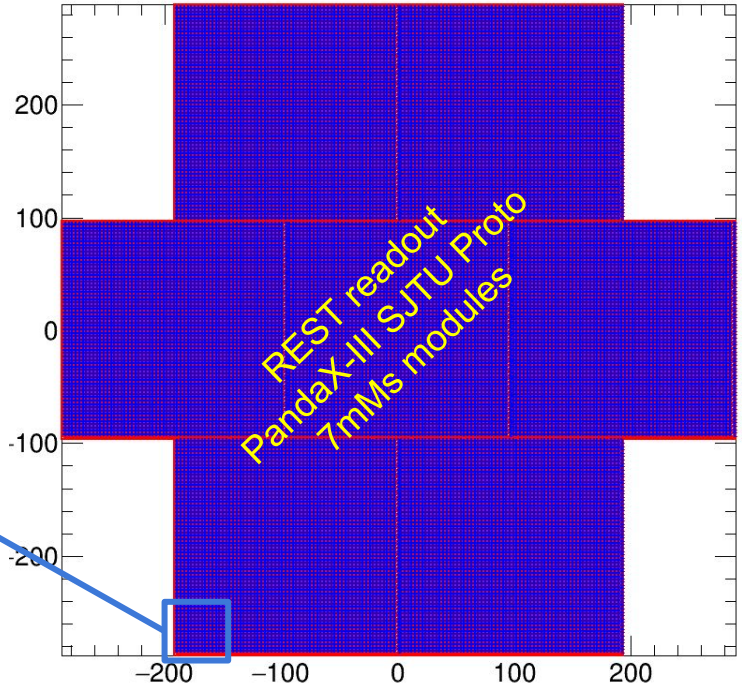
Gerber design from PandaX-III microbulk detectors



REST readout visualization



The readout is made of readout pixels - readout channels - readout modules



Gas mixture description. TRestMetadata::TRestGas

TRestGas is an interface to Garfield++/Magboltz

Example of precalculated gas mixtures described in RML

```
<section gas name="Gas-Xenon-TMA 1Pct" title="Xenon-TMA Mixture (1Pct TMA)">
    <parameter name="pressure" value="10" />
    <parameter name="temperature" value="293.15" />
    <parameter name="maxElectronEnergy" value="400" />
    <parameter name="ionizationPotential" value="10" />
    <parametere name="nCollisions" value="10" />
    <eField Emin="100" Emax="1000000." nodes="20" />
    <gasComponent name="xe" fraction="0.99" />
    <gasComponent name="n(ch3)3" fraction="0.01" />
</section>
```

Magboltz gas file generation can be launched in a ROOT interactive session.

*TRestGas *gas = new TRestGas("argonMixture.rml", "Argon-Isobutane 4Pct 10-10E3V/cm", true);*

Any gas available in Magboltz can be used in REST. The name convention can be found in Garfield++ user guide (Table B1).

http://garfieldpp.web.cern.ch/garfieldpp/documentation/UserGuide.pdf

restG4 package (simulation input from Geant4)

restG4 is a package allowing us to generate a first dataset in the REST event data and metadata format.

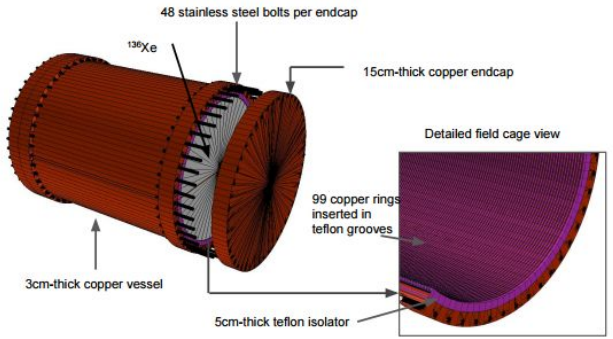
It requires :

- A description of the simulation conditions through an REST configuration file.

- A geometry definition in GDML compatible with ROOT.

Example with primaries definition : **U238** launched from volume **vessel** existing in GDML geometry

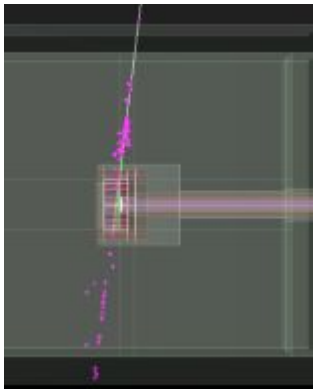
```
<parameter name="gdml_file" value="pandaXIII_Setup_WT12m.gdml"/>

<generator type="volume" from="vessel" >
    <source particle="U238" fullChain="on">
        <angularDist type="isotropic" />
        <energyDist type="mono" energy="0" units="keV" />
    </source>
</generator>
```



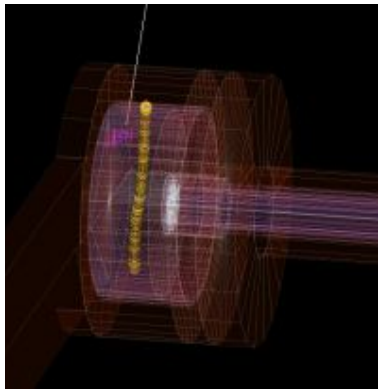
Examples of spatial and energy primary event distributions

```
<generator type="virtualWall" position="(0,400,0)" units="mm" size="200"
units="mm" rotation="(90.,0,0)">
   <source particle="mu-/gamma">
      <energyDist type="TH1D" file="../../inputData/distributions/Muons.root"
spctName="cosmicmuon" range="(0.1,350)" units="GeV" >
      <angularDist type="TH1D"
file="../../inputData/distributions/CosmicAngles.root" spctName="Theta2"/>
   </source>
</generator>
```

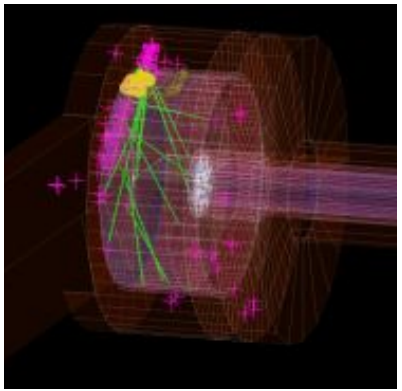
Cosmic muon

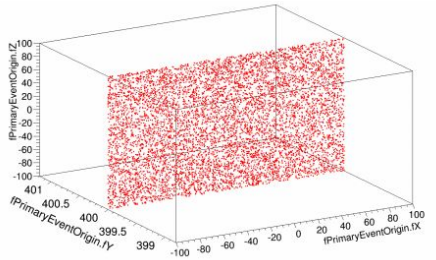


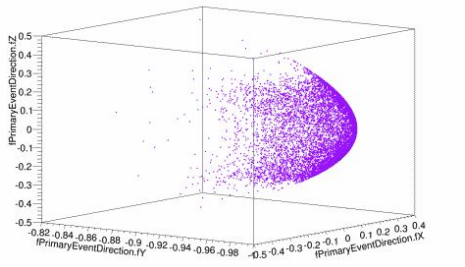
Low E cosmic Gamma (~MeV)



High E cosmic Gamma (~10GeV)



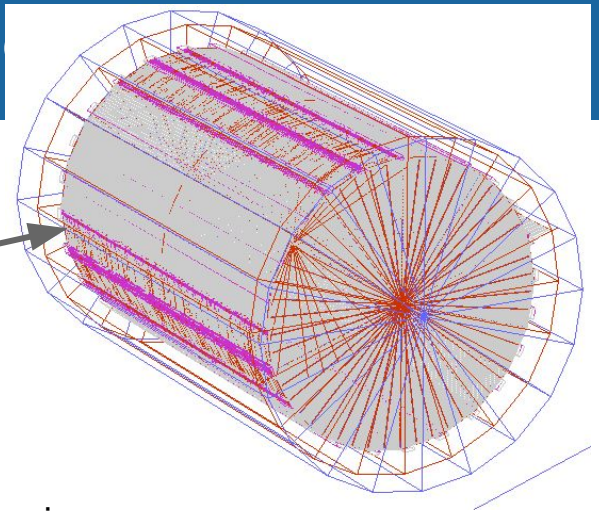




Simulations for IAXO detector by E. Ruiz Choliz/J. Gracia at Univ. of Zaragoza

GDML Geometry stored as TGeoManag

restG4 uses GDML geometry definition.



GDML in ROOT

Interactive root session:
TGeoManager *geo = new TGeoManager();
geo->Import("GeometrySetup.gdml");
geo->GetTopVolume()->Draw("ogl");

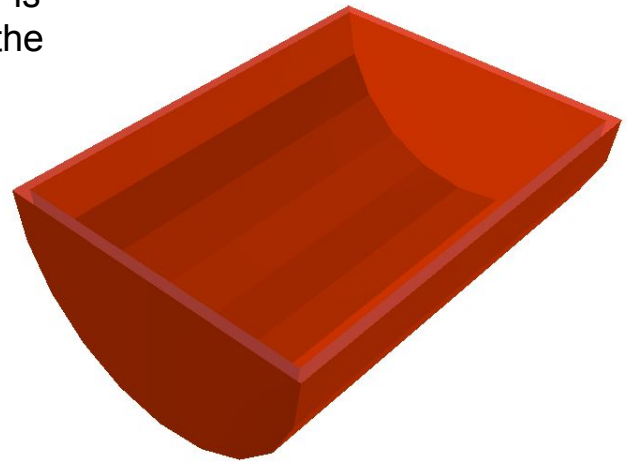
GDML Geometry in REST

Interactive root session:
TRestRun *run = new TRestRun();
run->OpenInputFile("myRESTFile.root");
TGeoManager *geo = run->GetGeometry();

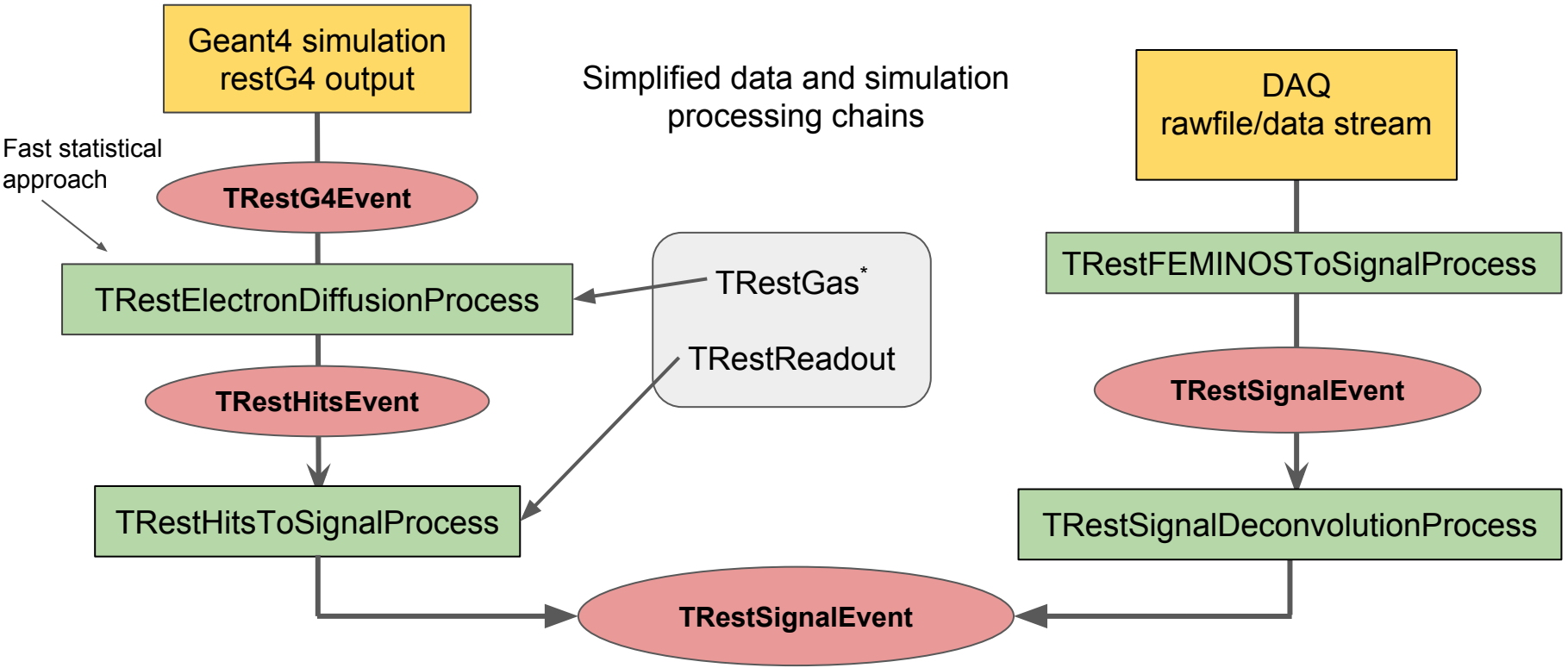
geo->GetVolume(3)->GetName();
>> (const char* 0x33343e9)"vesselVolume"

geo->GetVolume(3)->Draw("ogl");

The geometry is stored inside the REST file.



Towards comparison of rawdata and simulation data

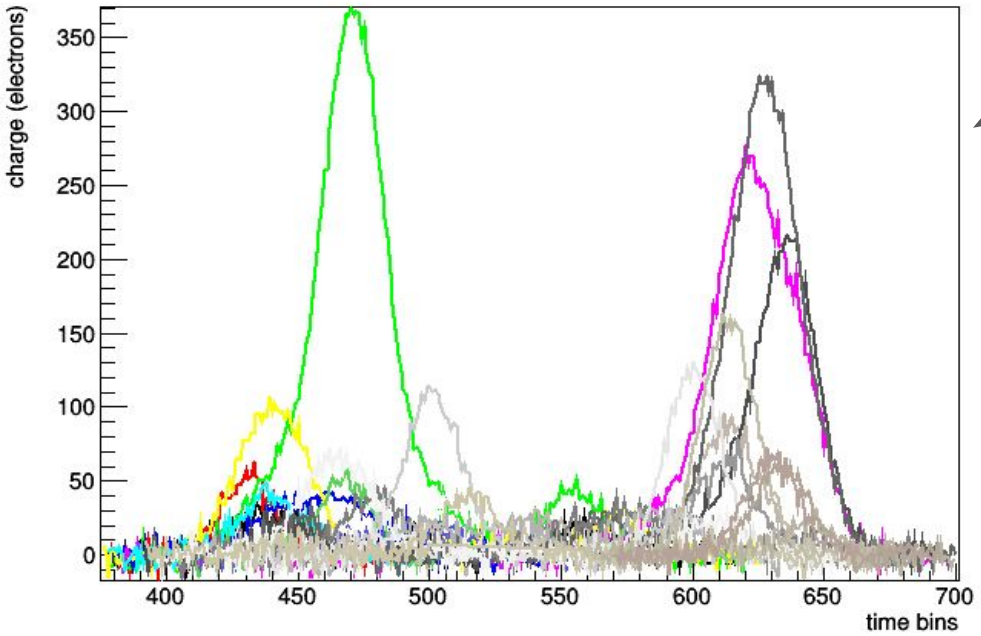


* Gas parameters as drift velocity or gas diffusion can be directly given to processes.
TRestGas allows to define the properties of any gas mixture by using the Garfield++ interface to Magboltz.

Daq event like simulation data

TRestElectronDiffusionProcess → TRestHitsToSignal → TRestSignalGaussianConvolution → TRestAddSignalNoise

Most simplified MC processing to get a reasonable detector signal response.

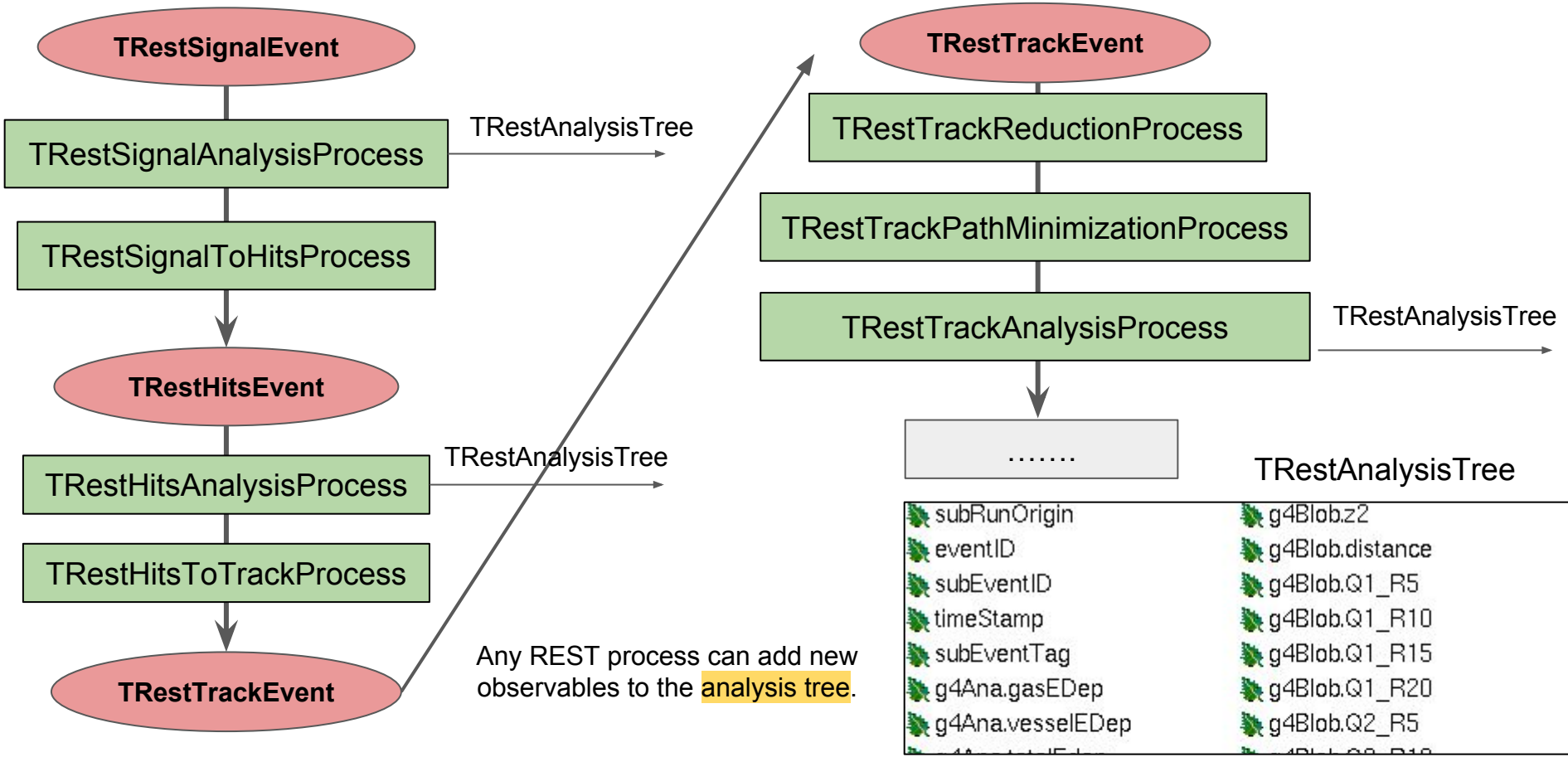


The main goal is to reach one point where the processing and analysis of events for **simulated data and real data is equivalent**.

We can also introduce an arbitrary signal response, measured or previously simulated.

TRestSignalConvolution

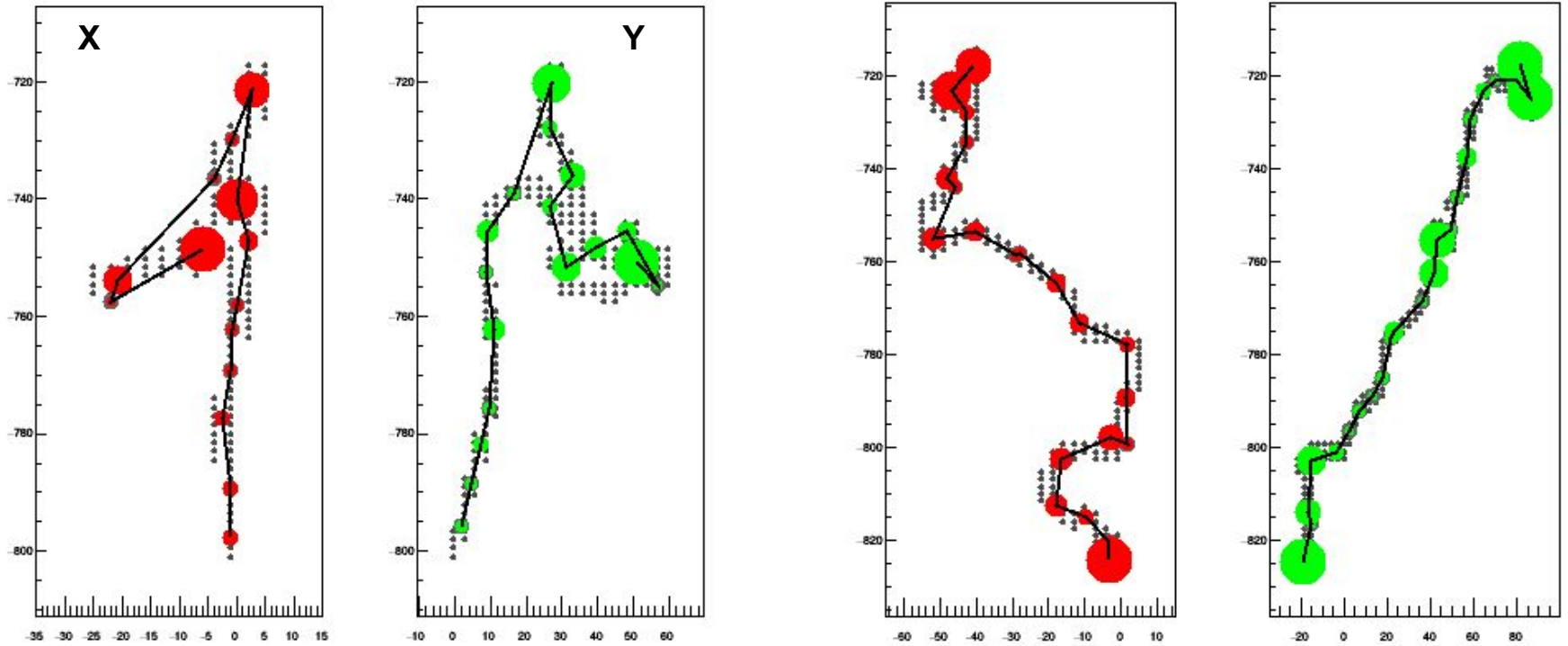
Common rawdata/simulation process chain



The resulting processed track event

XZ and YZ track event projections for simulated events using PandaX-III stripped readout.

Sampling : 1 us
Pitch : 3mm



TRestMetadata::TRestManager

The TRestManager class allows to define the main metadata information (readout, gas) and the processes we will apply into the input dat.

```
<section TRestManager name="CoBoToSignal" title="Converts to TRestSignalEvent and performs signal analysis" >

    <!-- +++++++++++++++
         Adding metadata
         +++++++++++++++ -->
        <!-- gas pressure units in TRestGas is atm -->
    <addMetadata type="TRestGas" name="Xenon-TMA 3Pct 10-10E3V/cm" pressure="1" value="ON" overwrite="false" >

    <addMetadata type="TRestReadout" name="SingleModule" value="ON" overwrite="false" >

    <!-- We modify/shift the readout plane existing in readout.root/SingleModule    -->

    <readoutPlane id="0" planePosition="(0,0,-255)" cathodePosition="(0,0,93)" />


    <!-- ++++++++++++++++
         Adding processes
         ++++++++++++++++ -->

    <!-- Signal analysis -->
    <addProcess type="TRestCoboAsadToSignalProcess" name="virtualDAQ" value="ON" >
    <addProcess type="TRestSignalAnalysisProcess" name="sgnlAna" value="ON" >

    <!-- Signal to track reconstruction and track analysis -->
    <addProcess type="TRestSignalToHitsProcess" name="signalToHits_Template" value="OFF" >
    <addProcess type="TRestFastHitsToTrackProcess" name="fastHitsToTrack-Template" value="OFF" >
    <addProcess type="TRestTrackAnalysisProcess" name="tckAna" value="OFF" >
```

List of processes → https://sultan.unizar.es/rest/

Can be found under tutorials page

Analysis processes

These are pure analysis processes. They do not transform the event data itself but add new observables/branches to TRestAnalysisTree. They may apply cuts to the event data, s being processed. Any process can decide to stop processing an event by just returning a NULL pointer. This should be documented in each process class.

REST process	Input type	Output type	Description
TRestRawSignalAnalysisProcess	TRestRawSignal	TRestRawSignal	Adds analysis observables from raw signal event.
TRestHitsAnalysisProcess	TRestHitsEvent	TRestHitsEvent	Adds analysis observables from hits event.
TRestGeant4AnalysisProcess	TRestG4Event	TRestG4Event	Adds analysis observables from a geant4 event.
TRestTrackAnalysisProcess	TRestTrackEvent	TRestTrackEvent	Adds analysis observables from a track event.
TRestTriggerAnalysisProcess	TRestSignalEvent	TRestSignalEvent	Applies cuts using time window and energy threshold trigger definition.
TRestFindG4BlobAnalysisProcess	TRestG4Event	TRestG4Event	Finds the electron end blobs in a TRestG4Event. For events with at least 2-electron tracks.

Signal processes

These processes just modify the data inside a signal event, returning again a signal event data type. These kind of processes add signal noise to simulated data, filter noise from ra data points which are under threshold.

REST process	Input type	Output type	Description
TRestAddSignalNoiseProcess	TRestSignalEvent	TRestSignalEvent	Adds random noise to a signal event.
TRestSignalDeconvolutionProcess	TRestSignalEvent	TRestSignalEvent	Deconvolutes a signal using a given input response signal.
TRestSignalGaussianConvolutionProcess	TRestSignalEvent	TRestSignalEvent	Convolutes the input signal with a gaussian.
TRestSignalShapingProcess	TRestSignalEvent	TRestSignalEvent	Shapes the input signal with a given input response signal.
TRestFindResponseSignalProcess	TRestSignalEvent	TRestSignalEvent	Selects clean signals from input to be used as response for deconvolution.
TRestSignalZeroSuppresionProcess	TRestRawSignalEvent	TRestSignalEvent	Keeps only points which are found over threshold.
TRestRawSignalRemoveChannelsProcess	TRestRawSignalEvent	TRestRawSignalEvent	Removes a number of selected channel ids from the input signal event.

REST encapsulated ROOT data format

REST files are self content and keep full traceability of the event data processing metadata.

METADATA + DATA always together full history and traceability

Typical REST file contents

```
TFile*         /home/javier/restData/Run_trackAnalysis_Xe136_00009_000(
 KEY: TRestG4Metadata  HM0. restG4;1   Xe136
 KEY: TRestTrackAnalysisProcess        P0. tckAna;1    Track analysis
 KEY: TRestGeant4AnalysisProcess       HP0. g4Ana;1    Geant4 analysis
 KEY: TRestFindG4BlobAnalysisProcess   HP1. g4Blob;1   Find blobs in D
 KEY: TRestG4toHitsEventProcess        HP2. G4ToHits;1 G4Hits to Hits
 KEY: TRestHitsShuffleProcess  HP3. hitsShuffle;1      Shuffles the hi
 KEY: TRestHitsToTrackProcess  HP4. hitsToTrack;1
 KEY: TRestTrackReductionProcess       HP5. trackReduction;1   Merges
 KEY: TRestTrackPathMinimizationProcess        HP6. trackPathMinimizat
 KEY: TRestTrackReconnectionProcess    HP7. trackReconnection;1
 KEY: TRestFindTrackBlobsProcess       HP8. tckBlob;1  Find blobs in t
 KEY: TTree    TRestTrackEventTree;1   Simulations PandaX-III v1.0
 KEY: TRestAnalysisTree        TRestAnalysisTree;1     Simulations Pan
 KEY: TGeoManager      Default;1       Geometry imported from GDML
 KEY: TRestRun PANDA-X III run;1       Simulations PandaX-III v1.0
root [4]
```

Geant4 simulation conditions

Processes metadata information

Event data

Analysis Tree

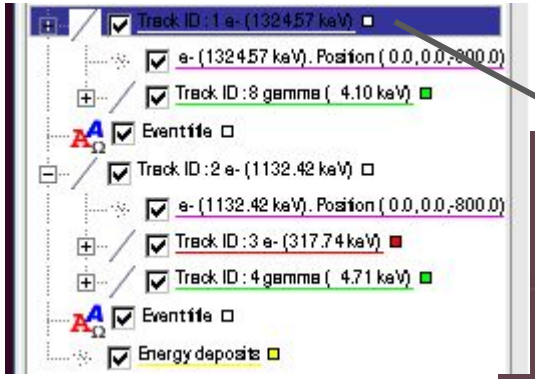
Geometry

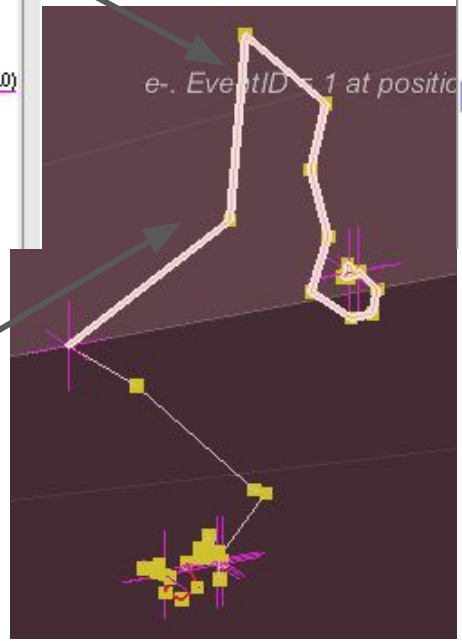
Run metadata

Event visualization using ROOT TEve classes

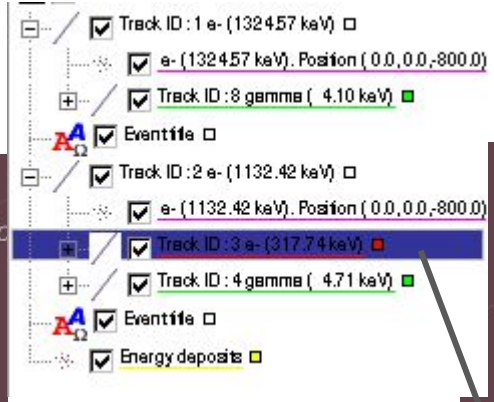
REST also provides visualizing tools for each basic event data type

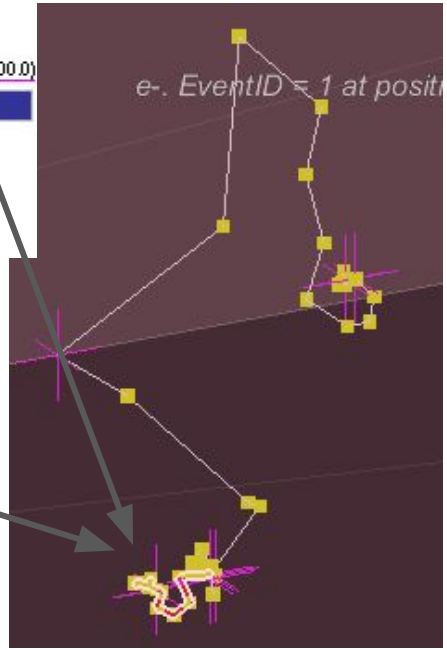
The Eve Manager facilitates the Geant4 event inspection and track identification.





Primary electron from DBD





Secondary electron produced by the second DBD electron track.

TRestAnalysisPlots (Systematic plot production)

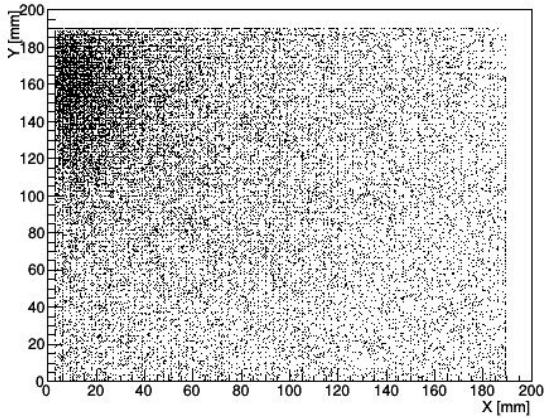
```
<plot name="Hitmap" title="Hitmap (from hitsAnalysis)" xlabel="X [mm]" ylabel="Y [mm]"
    logscale="false" save="/tmp/file3.png"  value="ON" >
    <variable name="hitsAna.yMean" range="(0,200)" nbins="1000" />
    <variable name="hitsAna.xMean" range="(0,200)" nbins="1000" />
</plot>
```

We can do 1D, 2D or 3D plots

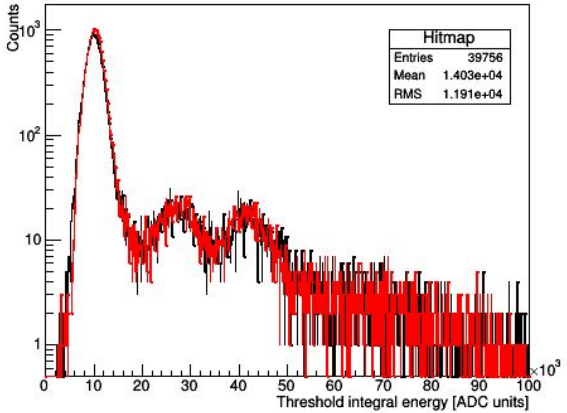
```
<plot name="Hitmap" title="Spectrum (single tracks)" xlabel="Threshold integral energy [ADC units]" ylabel="Counts"
    logscale="true" save="/tmp/file4.pdf"  value="ON" >
    <variable name="sgnlAna.ThresholdIntegral" range="(0,100000)" nbins="1000" />

    <cut variable="tckAna.nTracksX" condition="==1" value="ON" >
    <cut variable="tckAna.nTracksY" condition="==1" value="ON" >
</plot>
```

We can apply specific cuts to each plot definition

Hitmap (from hitsAnalysis)



Spectrum (single tracks)



TRestAnalysisPlots for quick look data production

TRestAnalysisPlot::
PlotCombinedCanvas()

It show a screen with all the plots defined in our plotAnalysis section.

<canvas size="(1000,800)" divide="(2,2)" />

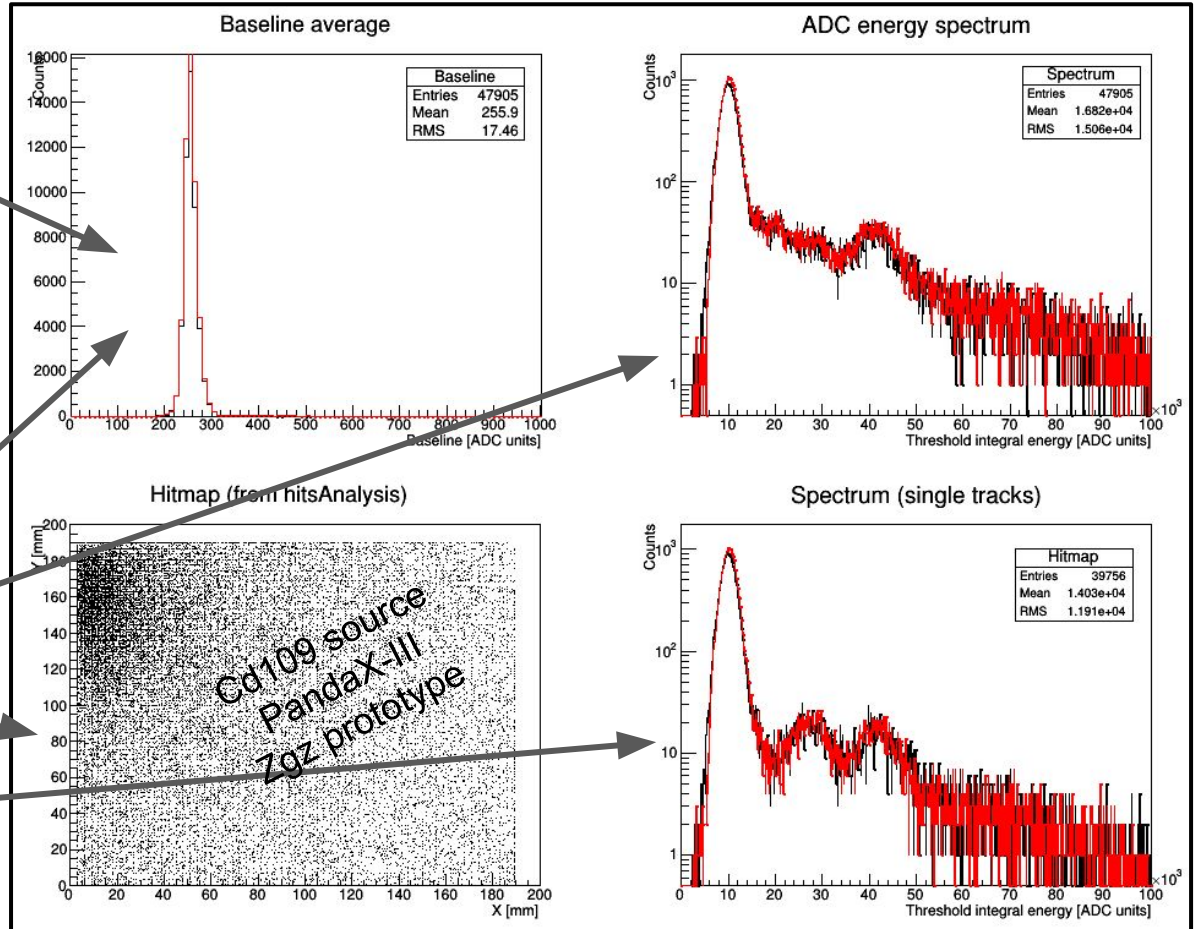
If the option save is enabled (we have provided a filename) each of the plots will be stored in the filename given.

<plot name="Baseline" …>

<plot name="Spectrum" …>

<plot name="Hitmap" …>

<plot name="Spectrum2" …>

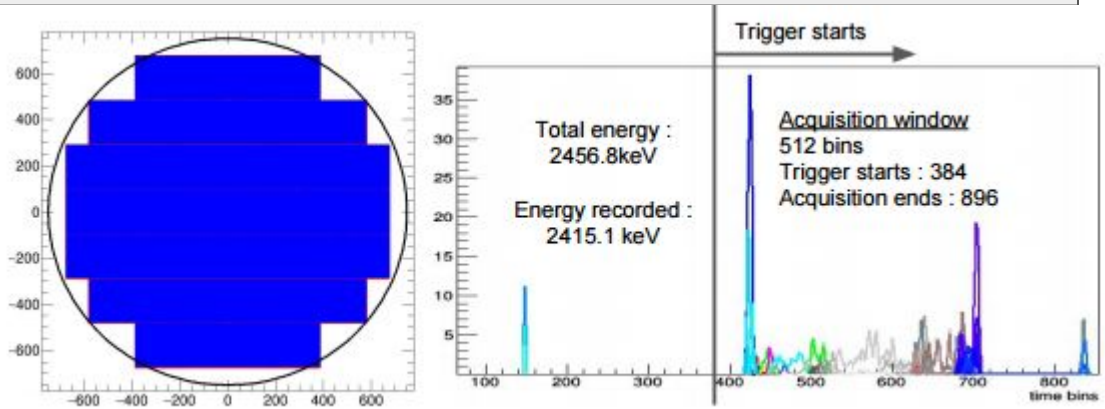


PandaX-III results produced with REST

The PandaX-III CDR was submitted to arXiv recently.

PandaX-III: Searching for Neutrinoless Double Beta Decay with High Pressure ^{136}Xe Gas Time Projection Chambers

https://arxiv.org/abs/1610.08883



Full event	Smearing (3% FWHM)	Time window	Active area
74.2%	70.8%	64.9%	54.2%

Background contribution and detector readout and time response was studied with existing REST processes and metadata structures.

Full study of different background contributions from different detector components.

Component	Isotope	Background (10^{-5} c/(keV·kg·y)) BambooMC	RestG4
Water	^{238}U	-	0.23
	^{232}Th	0.56	0.63
Barrel	^{238}U	1.07	2.41
	^{232}Th	7.54	7.86
	^{60}Co	3.02	2.11
End-caps	^{238}U	0.30	1.26
	^{232}Th	3.89	4.16
	^{60}Co	2.98	0.76
	^{238}U	3.50	11.9

Public documentation

https://sultan.unizar.es/rest/

RESTSoft v2.1 : Rare Event Searches with TPCs Software

Main Page | Namespaces | Class Documentation | Tutorials

RESTSoft v2.1 : Rare Event Searches with TPCs Software Documentation

Tutorials

Here it is a list of tutorials showing the basic usage of REST.

How to browse events using TRestRun	This tutor
How to create plot reports using TRestAnalysisPlot	This tutor
How to generate a gas file using TRestGas	This tutor
How to generate a TRestReadout structure	This tutor
How to generate Geant4 simulated data with restG4	This tutor
How to process data using TRestManager	This tutor
How to write your own REST processes library	This tutor
Installation of REST	Instructio
List of REST Processes	A list of th
List of REST Programs	A list of th

Some documentation available but still some effort needed to have a complete doc.

TRestG4Metadata Class Reference

Detailed Description

The main class to store the *Geant4* simulation conditions that will be used by *restG4*.

TRestG4Metadata is the main class used to interface with *restG4* (a REST based Geant4 co
information inside the output generated file. The simulations produced by *restG4* will write to
restG4 simulation and retrieve basic simulation results.

Summary

REST defines a data analysis framework with tools to study micropatterned readouts, specially designed for Rare Event Searches TPCs but easily transferred to other MPGD applications.

It fixes data format, metadata management and event data types. Allowing for data exchange, re-processing, and analysis comparison.

Scalability of the framework allows for event data process connectivity. Allowing the user framework to introduce his own specific processes into the framework.

New version of REST already being used to produce results in **PandaX-III**. Data/metadata layout processing ready for large data management.

Full framework interface through configuration files allows for quick learn curve making it ideal for new students.

Although code is not public but available on demand. Plans toward public version.