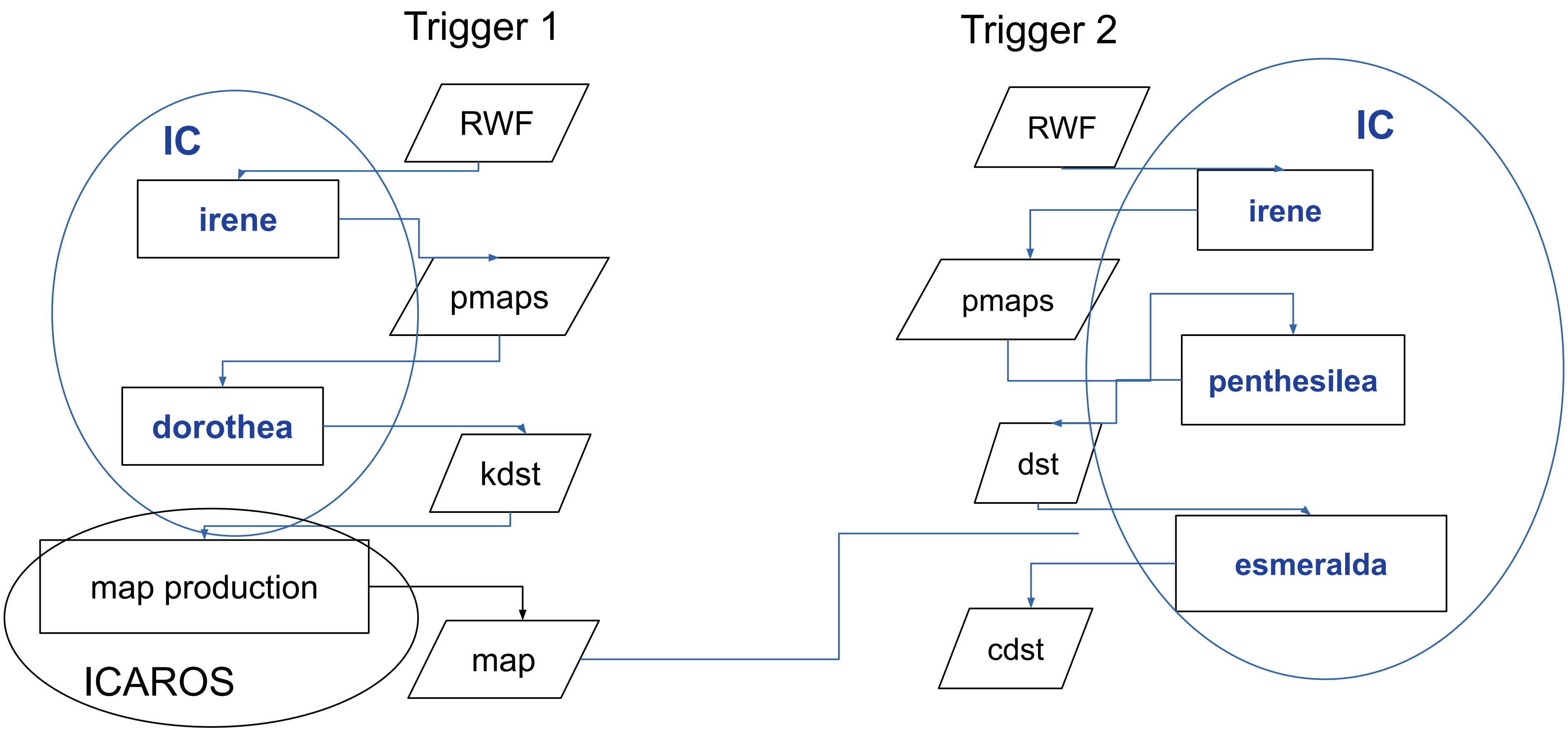
# Data processing algorithms

IC cities

# Official production flow



# Running a city method

After setting up the IC environment

source work_in_python_version 3.7

a city can be run as:

$ city city_name config_file.conf

# irene IO

**Input** (from the decoder or MC-detsim) – RWF (raw waveforms): a time-ordered signal amplitude for each sensor in ADC.

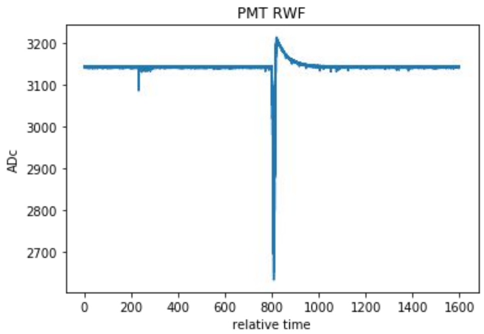
**Output** – pmaps (Peak maps): a collection of all the peaks and the slices of waveforms belonging to them.

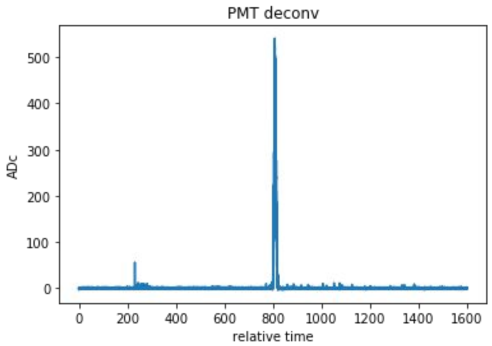
In RWF files the waveforms are in **RD/sipmrwf** and **RD/pmtrwf**.
Exercise: Using h5ls on rwf_example.h5 find the number of events and sensors inside the file. (h5ls -d will print the content of the table)

# irene algorithm

**1. PMT pedestal removed and subject to baseline restoration** (per pmt waveform)

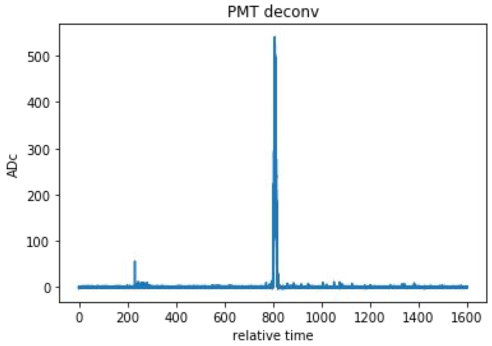
Pedestal - the average of the amplitude of the entire waveform;
Deconvolution using BLR algorithm (converts bipolar signal to monopolar)

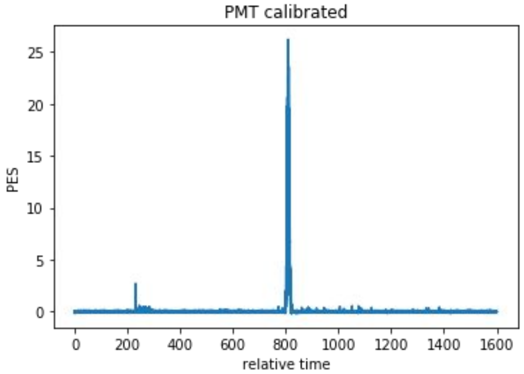


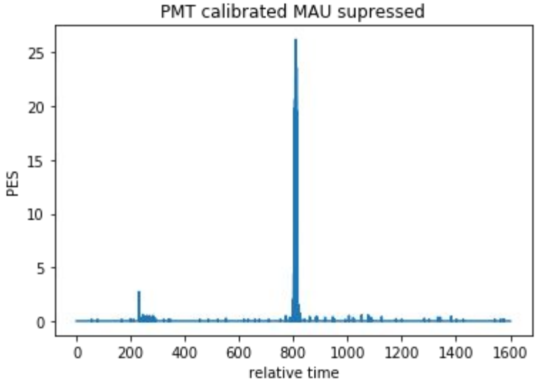



# irene algorithm

**2. correct wf** - ADC to pes (divide each waveform by the factor read from database)



# irene algorithm

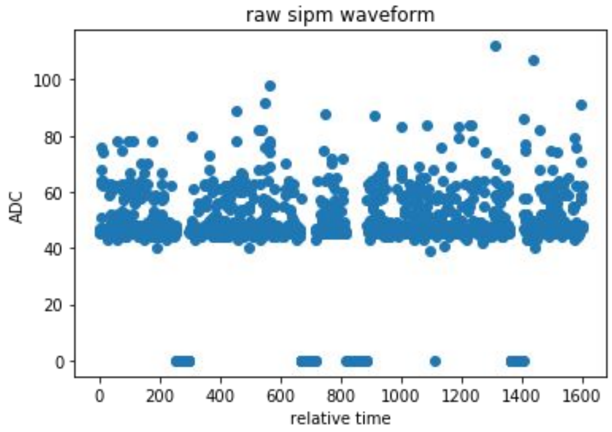
**3. zero suppressed** – find indices of bins where the **sum** of WFS is above given threshold. Two threshold (for S1 candidate use MAU suppressed corrected WF, for S2 candidates use corrected WF)

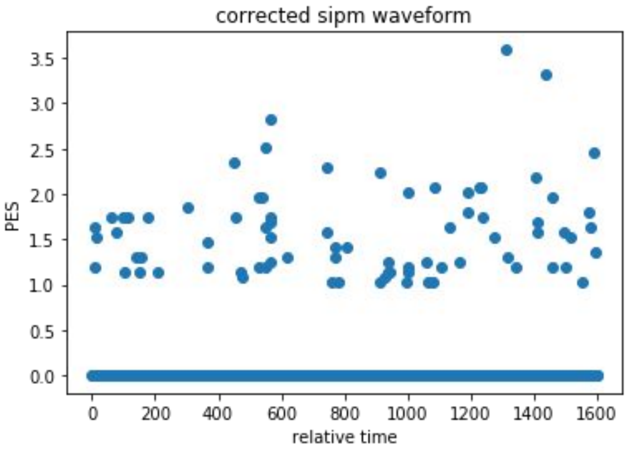
# irene algorithm

**4. correct and threshold sipm waveforms** – subtract baseline, convert to pes, set to 0 values below given threshold

Baseline is a mode of a waveform (the most frequent value)

Threshold is either ‘common’ (1 pes for all sipms), or ‘individual’ (percentage exclusion from calibration runs)



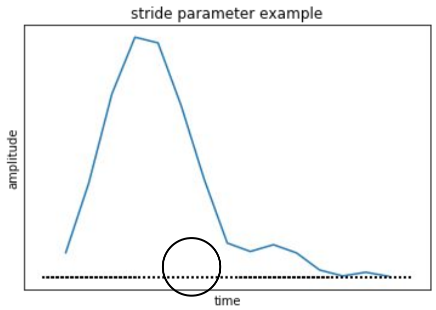

Note: sipm rwf are zero suppressed in FPGA

# irene algorithm

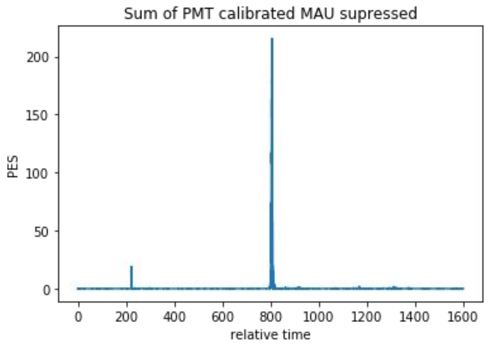
**5. pmap builder:** select s1/s2 peaks

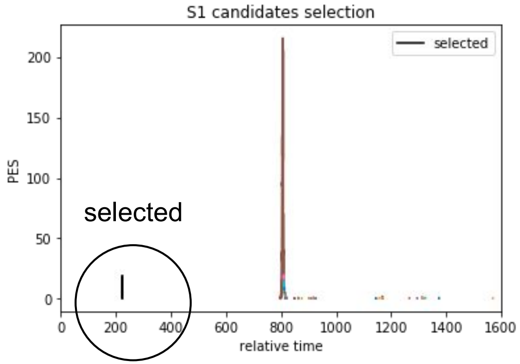
## S1 peak:

- split in disjoint signals (peaks) (using indices from step 3, allowing small gaps)
- search for peaks that satisfy time and length constraint




Value of ccwf goes below the threshold



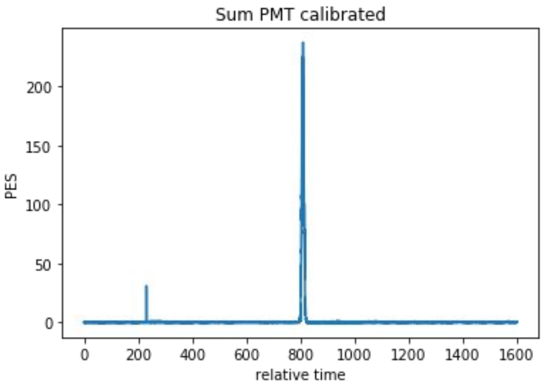



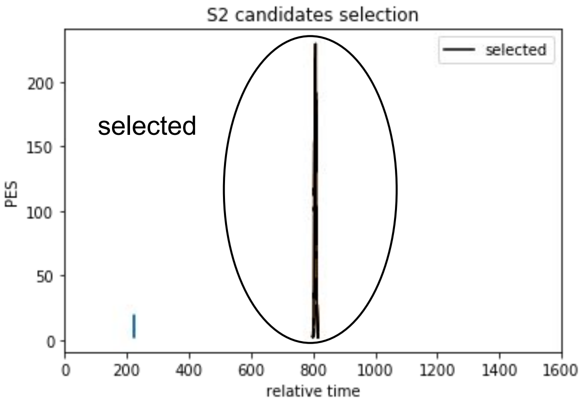
# irene algorithm

**5. pmap builder:** select s1/s2 peaks

**S2 peak:**

- split in disjoint signals (peaks)
- search for peaks that satisfy time and length constraint




- pmt signal is rebinned to match sipm binning (from 25 ns to 1 μs)
- include only sipms that have integrated (in time) signal larger than a given threshold (5pes)

# pmaps

In **PMAPS** tables are:

- **S1** – contains event, peak id, time, bin width, energy = summed wf over pmt for S1 peaks (per time bin) (summed over PMT waveform)
- **S1pmt** – contains event, peak id, pmt id , pmt signal (per pmt, per time bin)(waveforms)
- **S2** – contains event, peak id, time, bin width, energy = summed wf over pmt for S2 peaks(per time bin) (summed over pmt waveform)
- **S2pmt** – contains event, peak id, pmt id , pmt signal (per pmt, per time bin)(waveforms)
- **S2Si** – contains event, peak id, sipm id, sipm signal (per sipm, per time bin)(waveforms)

Exercise : run irene on rwf.h5 file using provided configuration.
Examine output file and find above-mentioned tables. What is the bin width for S1 and S2 peaks? How many S1 peaks were found in event 1046556? And S2? (Use head, tail or grep piping)

# dorothea IO

- **Input** (from irene) – pmaps
- **Output** – kdst (krypton dst): per peak relevant information (time, number of peaks, width, height, position...)

# dorothea algorithm

**1. select the peaks that satisfy a given set of conditions** on peak width, height and energy. The energy is calculated as a sum of bins that have amplitude greater than some en_th (usually 0pes for S2 and 0.5 pes for S1)

Events that contain no peaks with the given condition are filtered out

Note: energy (E) refers to signal detected on PMTs and charge (Q) to signal detected on SiPMs (que está mal pero bueno)

# dorothea algorithm

**2. build pointlike events**

per S1/S2 combination of peaks that passed selection calculate:

- for **S1** : width (in ns), height, energy (above threshold)
- for **S2** : width (in μs), height, energy  (above threshold), charge
- Extract xy position of sipms from the database. Find a 2D barycenter of the whole S2 peak (i.e. integrated over peak duration)
- Find Z=1 * (S2_time_at_max_energy-S1_time_at_max_energy) (in mm) (Note that drift velocity is just assumed to be 1 mm/μs)

Note: peak_id is resetted and does not correspond to pmaps peak_id!

# kdst

**DST/Events** table contains :

event, time, s1_peak, s2_peak, nS1, nS2,
S1w, S1h, S1e, S1t, S2w, S2h, S2e, S2q, S2t,
Nsipm, DT, Z, Zrms, X, Y, R, Phi, Xrms, Yrms

Exercise: run dorothea on the previously produced pmaps using provided config file.
Is there any event with more than one S1 and one S2?

# penthesilea IO

- **Input** (from irene) – pmaps
- **Output** – hdst (hit dst): per energy deposition (hit) relevant information (position, charge, energy...)

# penthesilea algorithm

**1. select peaks that satisfy a given set of conditions** on peak width, height and energy. The energy is calculated as a sum of bins that have amplitude greater than some en_th (usually 0pes for S2 and 0.5 pes for S1)

Keep events that have one and only 1 S1 (standard configuration but not constrained by the code)

# penthesilea algorithm

**2. build hits**

- rebin S2 waveform (usually to bins of 2 μs width)
- take S1_time_at_max_energy of first S1 peak as beginning time
- per S2 peak:
  - Extract xy the position of the SiPMs from the database for all SiPMs in the peak per time bin:
    - runs reco algorithm (finds X, Y, Q position of a hit)
    - distribute the energy (E) of the bin to hits found
    - If no hit was found, the X, Y position set to 0, Q to NN (-99999)

The result is a collection of X, Y, Z, E, Q … variables per time bin

# penthesilea algorithm

**3. create kdst like table**

- The same procedure as in dorothea.

# reco algorithm (corona)

1) Ignores sipms with charge below Qth;
2) Select hottest sipm with charge above Qlm;
3) Finds first barycenter inside lm_radius around the hottest sipm;
4) Finds second barycenter in new_lm_radius around first barycenter
5) If there are at least msipm sensors the hit is accepted

To set 1 hit per sipm reco paradigm (used for hdst):

```
Qthr  =  5 * pes
Qlm  =  5 * pes
lm_radius =  0 * mm
new_lm_radius   =  0 * mm
msipm = 1
```

To set global barycenter (used for kdst):

```
Qthr  =  1 * pes
Qlm  =  0 * pes
lm_radius =  -1 * mm
new_lm_radius   =  -1 * mm
msipm = 1
```

1 hit per sipm means that the XY position of the hit will always correspond to an XY position of the SiPMs

# hdst

**RECO/Events** table:

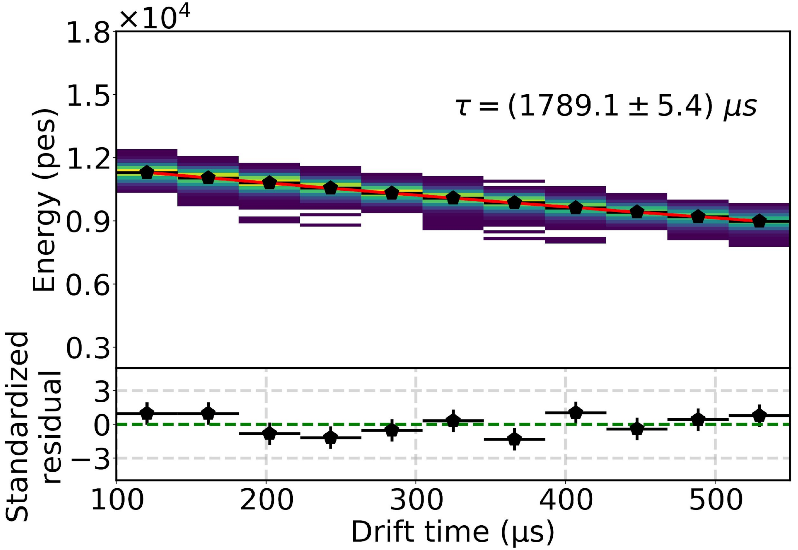
event, time, npeak, Xpeak, Ypeak, nsipm, X, Y, Xrms, Yrms, Z, Q, E, (Qc, Ec , track_id, Ep set to -1)

Exercise:
run penthesilea on pmaps.h5 using provided config file.
What is the charge of the first hit of the event 1046556? Why? What’s its energy?

# map production

1) filter dst to select good kr events
2) bin detector in xy (ie you get n_bins x n_bins arrays of E, Z)
3) per bin fit E, Z to

$$E_{\mathrm{measured}} = E_{\mathrm{true}} \times E0 \times \mathrm{e}^{-Z_{\mathrm{measured}/\mathrm{LT}}}$$




4) Store 2d maps of E0 and LT

# esmeralda IO

- **Input** (from penthesilea) – hdst
  (from map_production) - maps

- **Output** – cdst (corrected hit dst): the same as hdst with additional energy correction
  track information : results of running paolina analysis

# esmeralda algorithm

1. correct hits:

  - Apply a new charge threshold to the hits. The energy of the hits that don’t pass this threshold is redistributed among the hits that have the same time (Z)
  - get rid of failed (NN) hits : energy of those hits is redistributed to the (in Z) closest hits
  - using correction map calculate corrected energy (the units are keV at Kr scale)

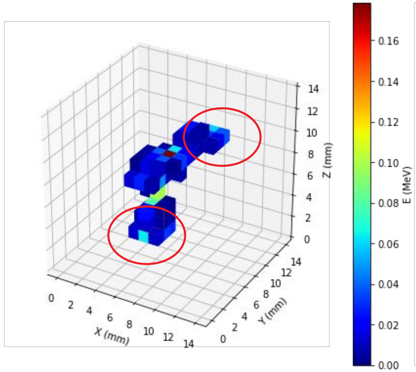
$$E_{\mathrm{corrected}} = E_{\mathrm{measured}}/E0 \times \mathrm{e}^{\,Z_{\mathrm{measured}/\mathrm{LT}}}$$

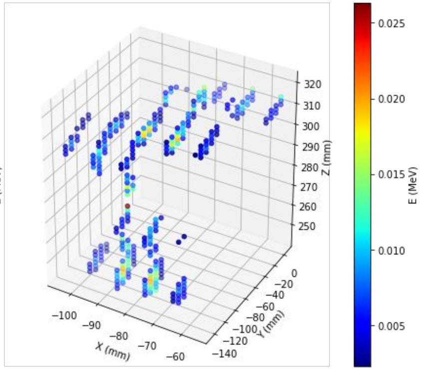
This function is run two times:

1. for lower threshold used in deconvolution and DNN analysis (5-10pes)
2. for higher threshold used for everything else (30-35pes)

# esmeralda algorithm

2. run paolina functions (finding voxels and tracks for higher thresholded hits):
   - voxelize event
   - find tracks
   - drop end_point voxel if its energy is smaller than a given threshold, redistribute its energy
   - find blobs (energy deposition at the end of the tracks)



# cdst

- **RECO/highTh** and **RECO/lowTh**
  The output of correction procedure, has the same structure as the penthesilea output.
  (Qc, Ec, track_id, Ep have meaningful values)

- **Tracking/Tracks** is per track information containing :
  event,  trackID, energy, length, numb_of_voxels, numb_of_hits, numb_of_tracks,
  x_min, y_min, z_min, r_min, x_max, y_max, z_max, r_max, x_ave, y_ave, z_ave, r_ave,
  extreme1_x, extreme1_y, extreme1_z, extreme2_x, extreme2_y, extreme2_z,
  blob1_x, blob1_y, blob1_z, blob2_x, blob2_y, blob2_z, eblob1, eblob2,
  ovlp_blob_energy (energy of the hits belonging to both end point blobs),
  vox_size_x, vox_size_y, vox_size_z

- **Summary/Events** table containing per event information:
  event, evt_energy, evt_charge, evt_ntrks, evt_nhits, evt_x_avg, evt_y_avg, evt_z_avg, evt_r_avg,
  evt_x_min, evt_y_min, evt_z_min, evt_r_min, evt_x_max, evt_y_max, evt_z_max, evt_r_max,
  evt_out_of_map (the XYZ position of event is outside detector limits?; if positive energy and average positions will be nans)

- **DST/Events** – copied table from penthesilea output

# cdst

Exercise: run esmeralda. What is the energy of the first track of event 1046556,?
Which energy has the highest energy? Which table has more entries: lowTh or highTh? Why?

# Questions?