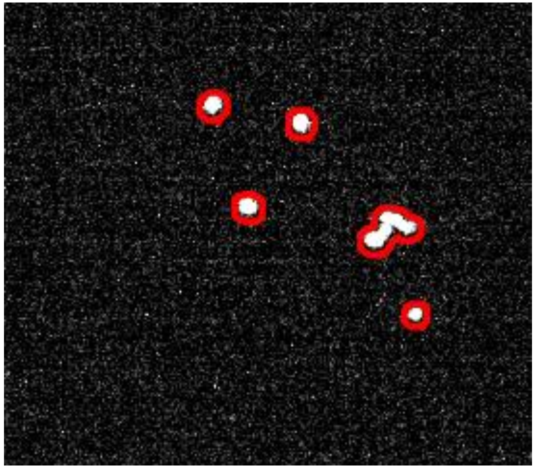
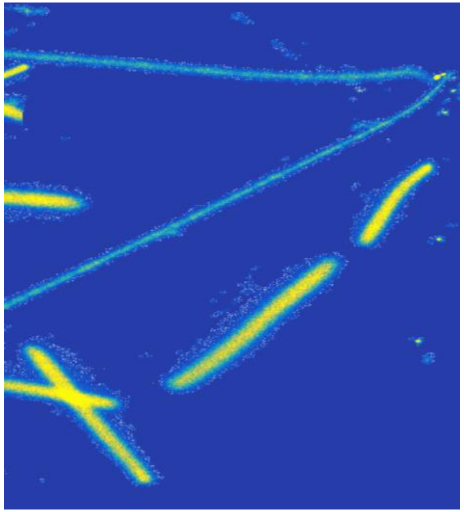


Status of Reconstruction code

Giorgio Dho

Istituto Nazionale di Fisica Nucleare (INFN-LNF), Frascati (RM), Italy

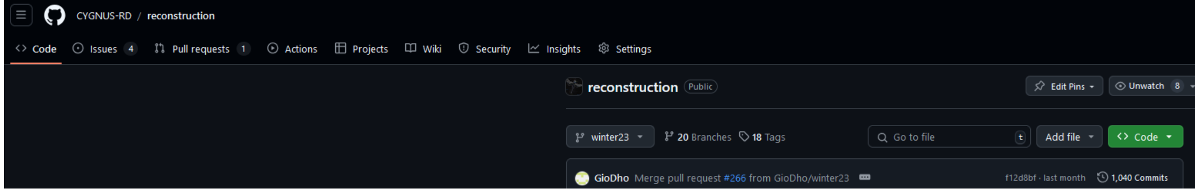


CYGNO Collaboration meeting

G. Dho, Cagli

Nov 28 2024

Reconstruction Flow



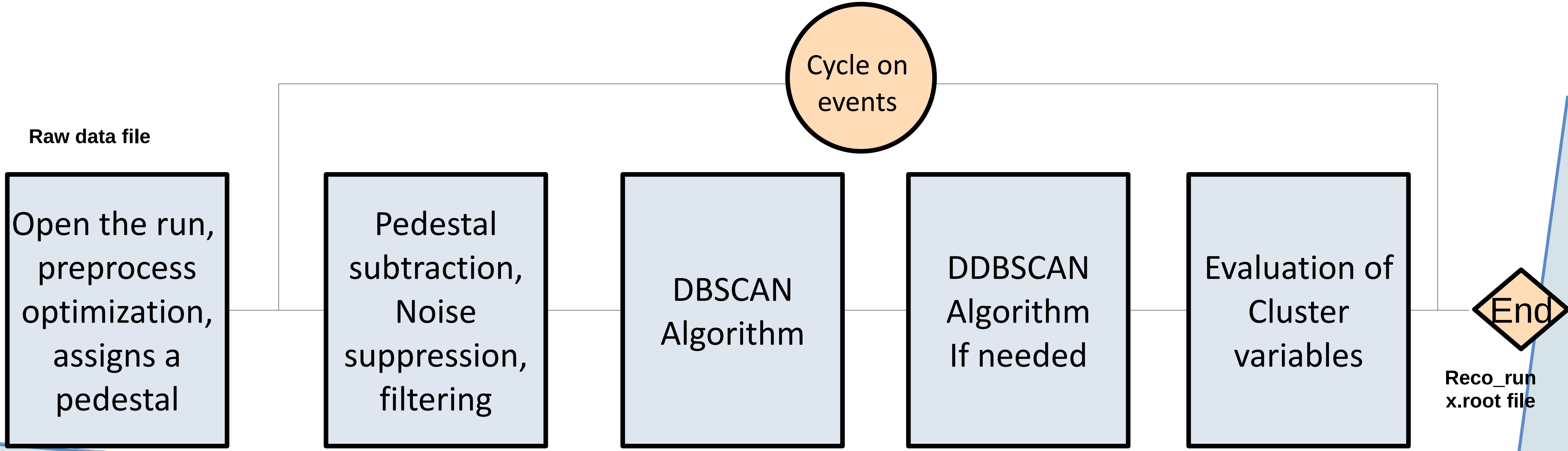
GioDho Merge pull request #266 from GioDho/winter23 f12d8bf · last month 1,040 Commits

- The current more updated branch is still **Winter23**.
- Now also releases are available with stable enough operation (William Wallace, JoanD’Arc.. Saladin soon to join)

Comparing the same screenshot of 1 year ago:

35 PR merged

176 new commits


Raw data file
Cycle on events
Open the run, preprocess optimization, assigns a pedestal
Pedestal subtraction, Noise suppression, filtering
DBSCAN Algorithm
DDBSCAN Algorithm If needed
Evaluation of Cluster variables
End
Reco_run x.root file

One year ago

- Last year various items to tackle were shown

NOW		THEN		Future
Parameters for Run3 and Run4	☑	Pedestal storage	not needed	Modular Reco
What to do with sensor borders	☑	Adapt QUEST	☑	Partially integrated in DAQ
Saving parameters	☑	Postreco tools	ongoing	
Rotation correction	☑	Uproot issue	☑	
Include PMT	☑			

Special thanks

here to Rafael

Stability and fixes

- Discord channel and PR method proved to be a very effective tool

Hopefully I was responsive a quick enough with who had reco problems

No indirect crashes introduced in LIME reconstruction by PRs

- Most of the fixes were about:

Better handle of multithread

More checks on environmental variables to avoid crashes

Addition of online databases to look up to speed up (especially debug mode)

Noise filtering cythonized

Retrocompatibility (ROOT files) and tools for MC analysis

Timestamp in microseconds added (since October 7th) to better match PMT signal to camera images

Milestones achieved

- Biggest additions of the year:

PMT inclusion in the reconstruction code	**Orientation of the camera data**	**QUEST camera analysis unlocked**
Since 28th February	Since 21st May	Since 21st May

PMT as part of the RECO

Special thanks to

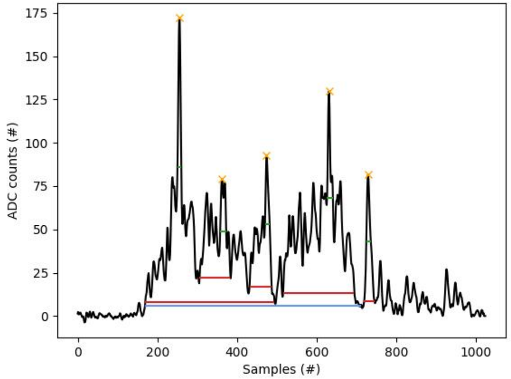
David and Pietro

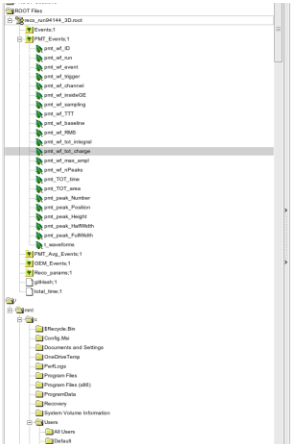
- The PMT waveform analysis is officially part of the reco code and can be toggled from ON to OFF by the user

- Even though officially released at the end of February, all Run3 and Run4 data are fully analysed with PMT reco ON

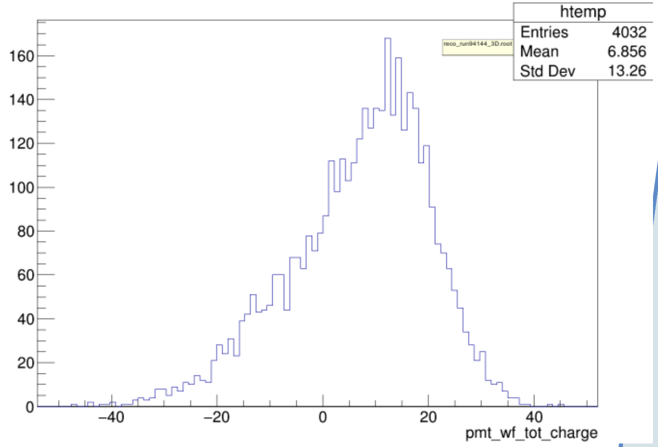
- PMT reco provides basic peak findings, time over threshold, charge, estimation of charge duration for all waveforms and for the average version of the 4 PMTs for each event. All embedded in the output reco file

- ReadME available in reconstruction folder



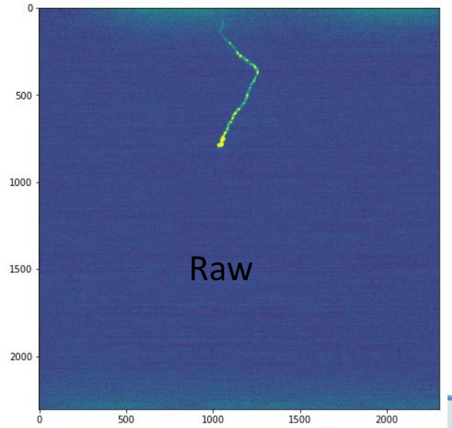


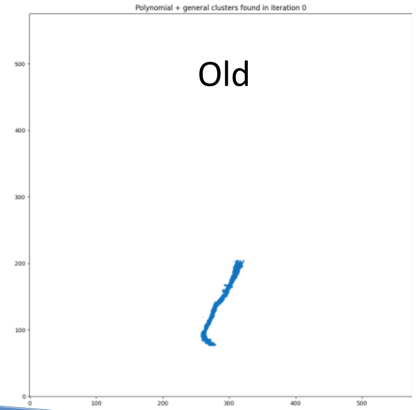
```
ROOT Files
⊟ reco_run94144_3D.root
    Events;1
  ⊟ PMT_Events;1
      pmt_wf_ID
      pmt_wf_run
      pmt_wf_event
      pmt_wf_trigger
      pmt_wf_channel
      pmt_wf_insideGE
      pmt_wf_sampling
      pmt_wf_TTT
      pmt_wf_baseline
      pmt_wf_RMS
      pmt_wf_tot_integral
      pmt_wf_tot_charge
      pmt_wf_max_ampl
      pmt_wf_nPeaks
      pmt_TOT_time
      pmt_TOT_area
      pmt_peak_Number
      pmt_peak_Position
      pmt_peak_Height
      pmt_peak_HalfWidth
      pmt_peak_FullWidth
      t_waveforms
    PMT_Avg_Events;1
    GEM_Events;1
    Reco_params;1
    gitHash;1
    total_time;1
/
⊟ mnt
  ⊟ c
      $Recycle.Bin
      Config.Msi
      Documents and Settings
      OneDriveTemp
      PerfLogs
      Program Files
      Program Files (x86)
      ProgramData
      Recovery
      System Volume Information
    ⊟ Users
        All Users
        Default
```

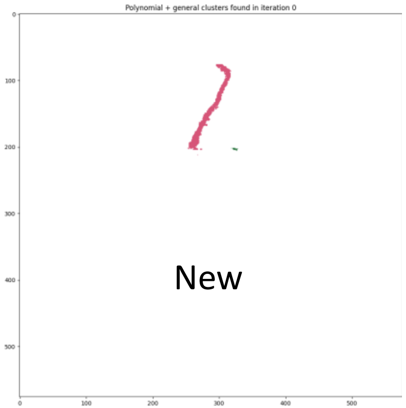


Orientation of the tracks

- As a directional detector we need to know how oriented the data is
- Weird transposition, rotation and UNO reverse were present
- To be remembered: row and columns are y and x; the (0,0) coordinate being on the top left of the image
- All data **from beginning of Run5** are correctly oriented





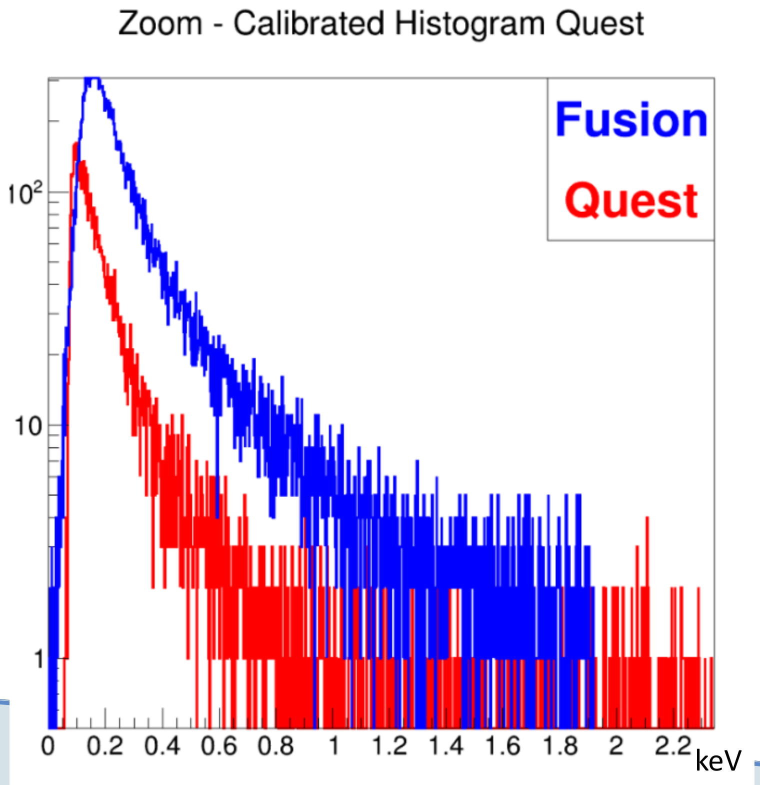


QUEST Introduction

- ORCA QUEST will be the CYGNO-04 camera and it has already been used for polarimetry studies

- It required to tinker with geomentry (not a square sensor anymore) and different counting of the camera

- This is now fully working

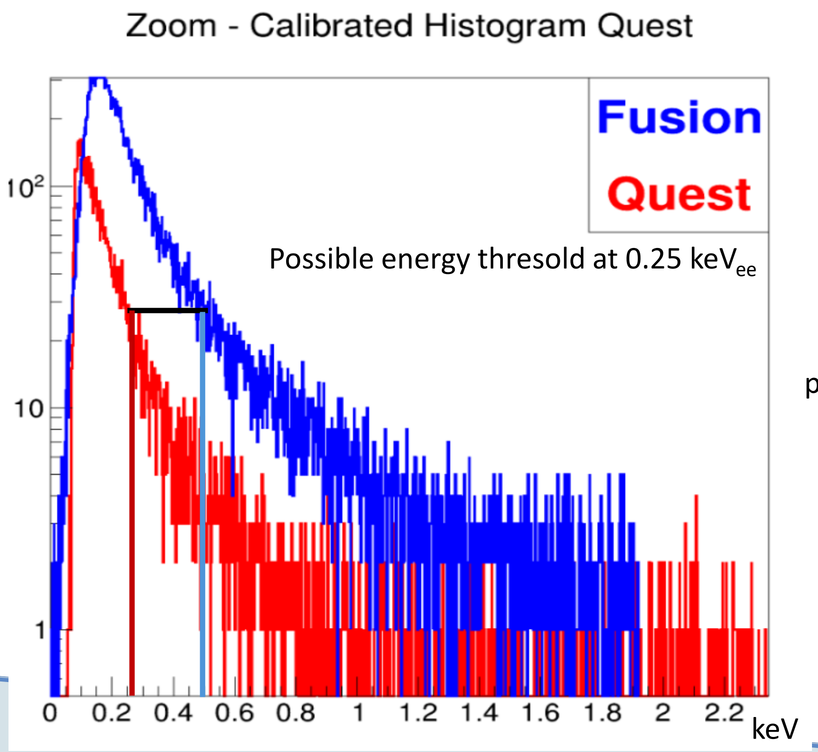


QUEST Introduction

- ORCA QUEST will be the CYGNO-04 camera and it has already been used for polarimetry studies

- It required to tinker with geomentry (not a square sensor anymore) and different counting of the camera

- This is now fully working



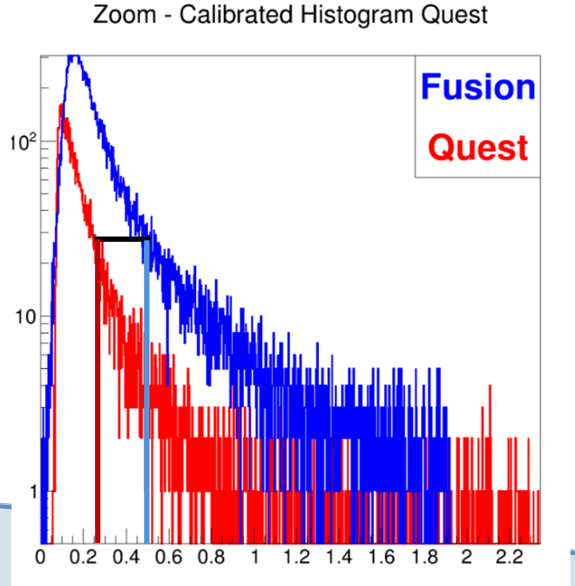
Zoom - Calibrated Histogram Quest

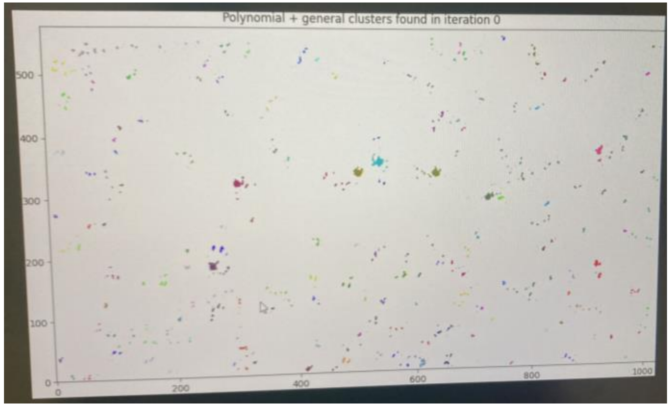
Possible energy thresold at 0.25 keV$_{ee}$

0.5 keV$_{ee}$ suggested by Stefano's presentation during November in person meeting

QUEST Introduction

- ORCA QUEST will be the CYGNO-04 camera and it has already been used for polarimetry studies
- It required to tinker with geomentry (not a square sensor anymore) and different counting of the camera
- This is now fully working





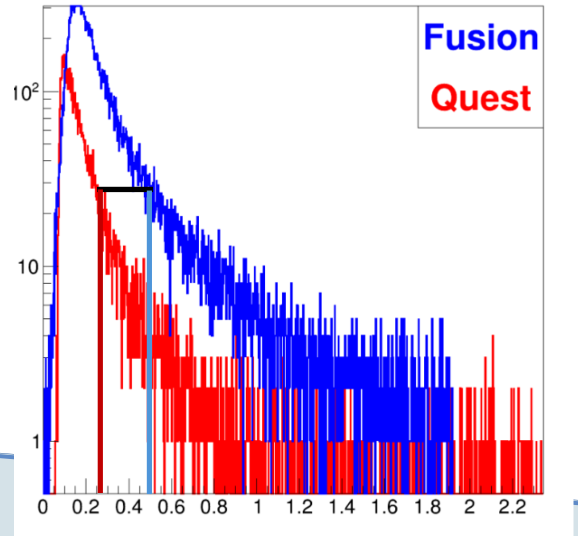
QUEST Introduction

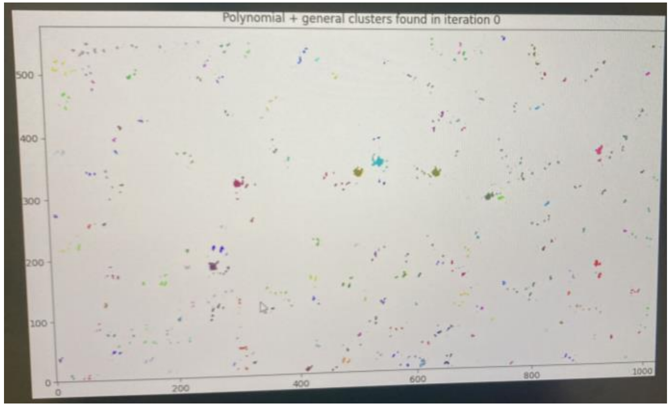
- ORCA QUEST will be the CYGNO-04 camera and it has already been used for polarimetry studies

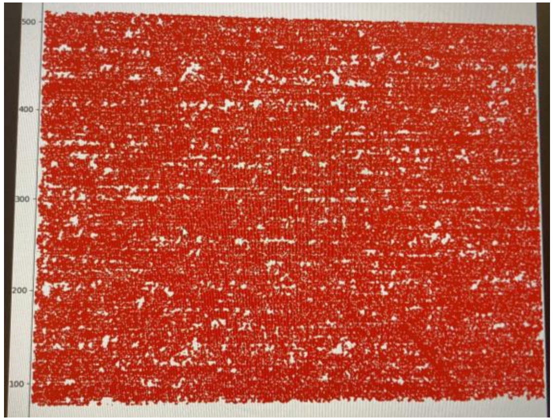
- It required to tinker with geomentry (not a square sensor anymore) and different counting of the camera

- This is now fully working

Zoom - Calibrated Histogram Quest







Vignetting production

- With new lenses being planned for CYGNO-04, new vignetting maps need to be produced
- Considering also different prototypes, change of lens, variation in aperture or spacers, and possibly distance to the object can modify the vignette response
- New small code developed which takes an image of a white surface and returns the vignetting map
- Vignette maps can be found stored in the cloud (description in ReadME present in the reconstruction repository)
- **Establishment of new methodology:** Every new optical system (lens, aperture, distance to object) needs to image and save a picture of a white surface nicely illuminated to have the vignette computed (for CYGNO-04, this will need to be done in advance of the data taking)

Possible input formats:

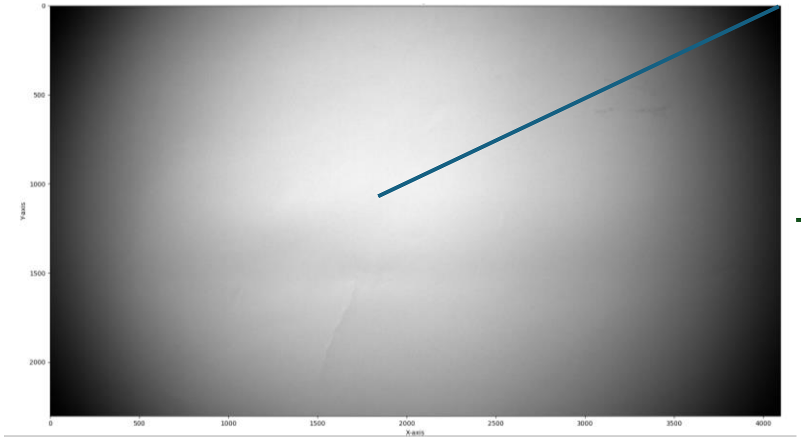
midas file

.dat file saved from Hokawo

Vignetting production

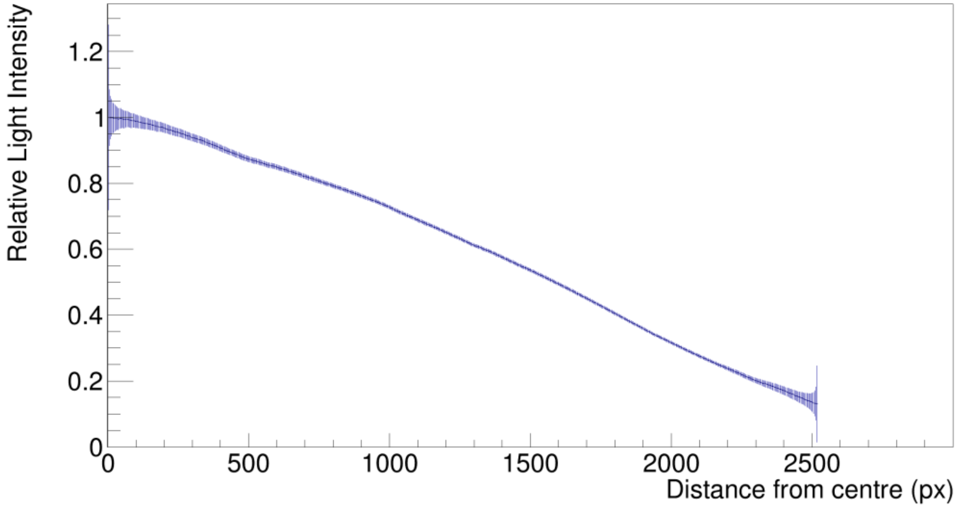
- The code finds the more luminous point and calculates the light decrease for every pixel

- Then the vignette is centered and the 2D map calculated



The more uniformly illuminated

and flatter, the better

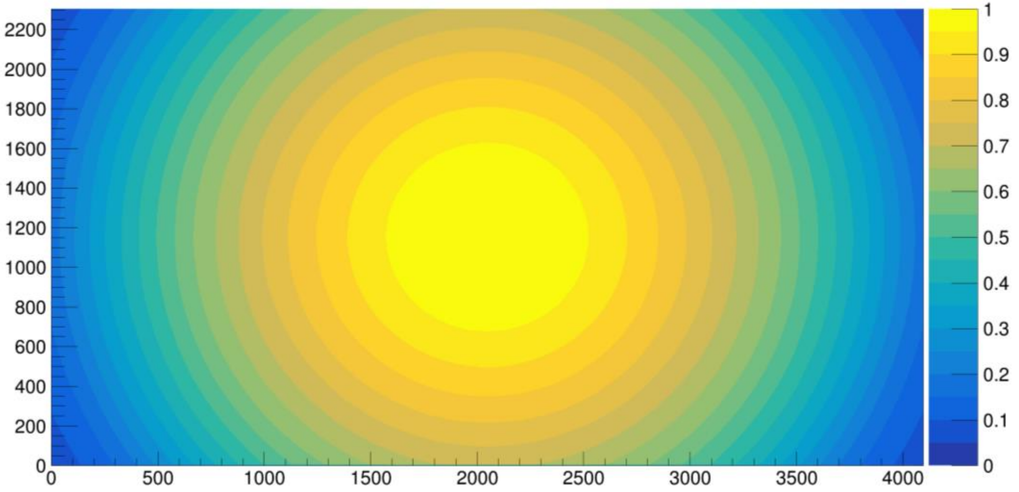


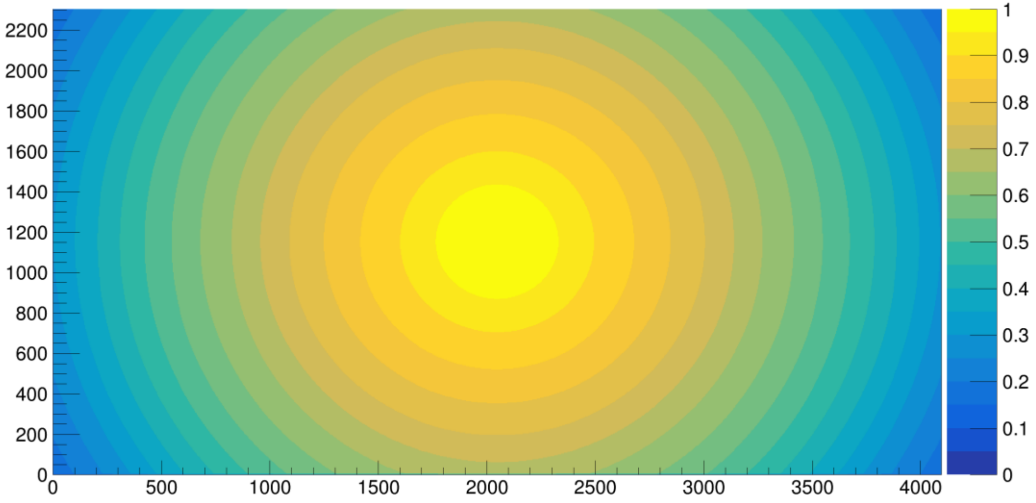
Reflected light or emission light (LED plan)

still unknown

Vignetting production

- From the 1D the 2D can be easily produced

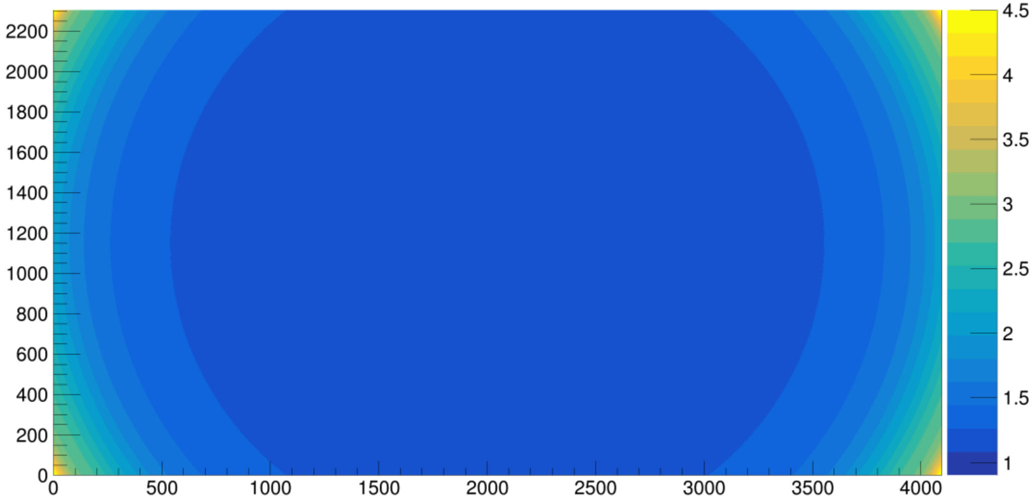

Quest Xenon lens 0.95 aperture

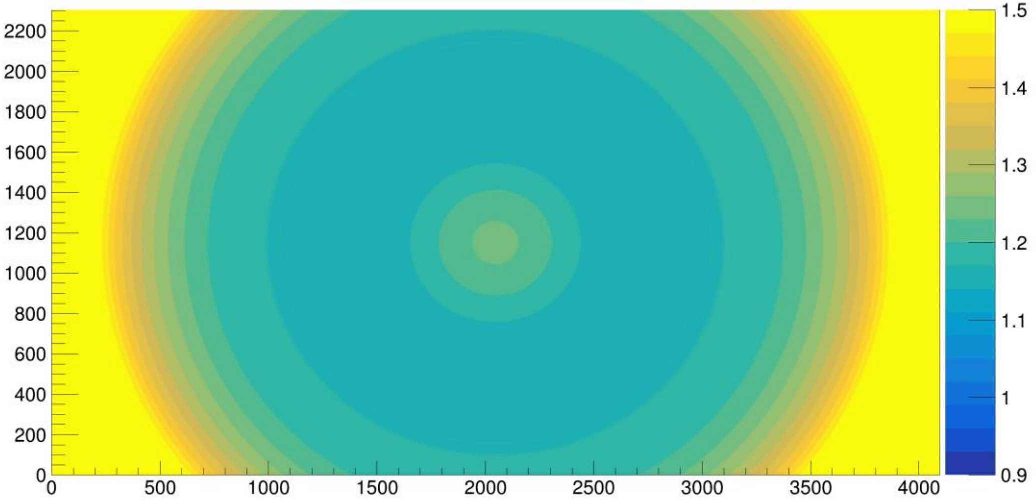

Quest EHD lens 0.85 aperture

EHD seems to improve light on side bands by more than factor 2

Vignetting production

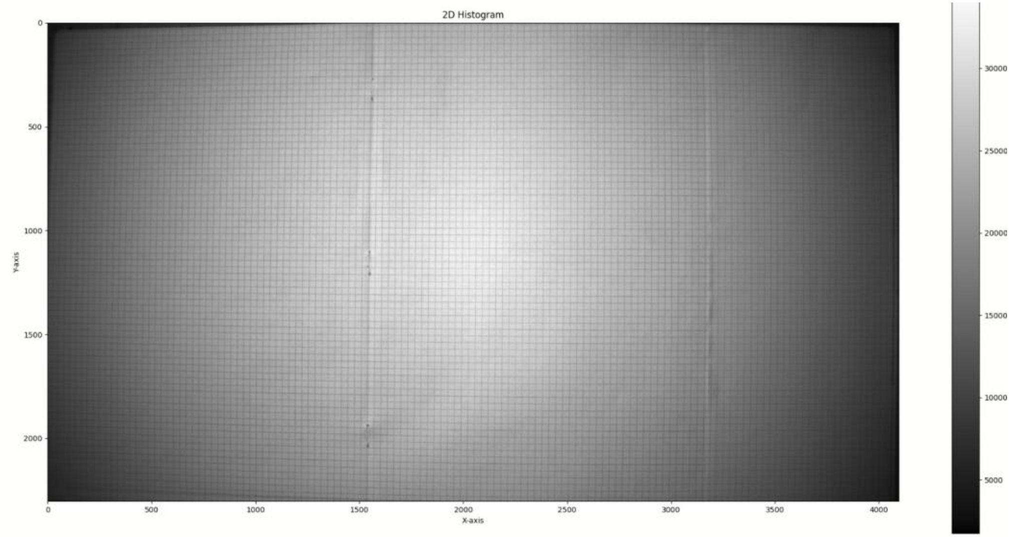
- The EHD with 0.85 aperture should grant 25% more light than Xenon lens (solid angle formula)
- Ratio of the two vignette maps (EHD 25% more light divided by Xenon)

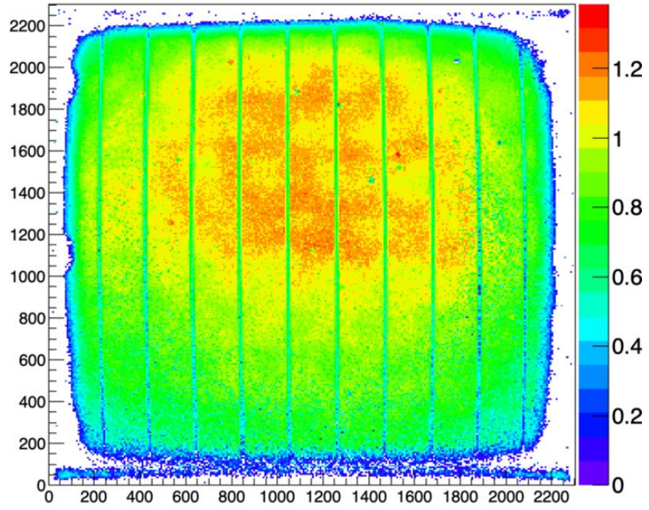




Barrel correction

- Lens with wide aperture results in distortion which bulges outward the image





- This distortion can be corrected with a change in the coordinates.

- In our case a 5 degree polinomial should do the trick, after clusterization

Barrel Correction

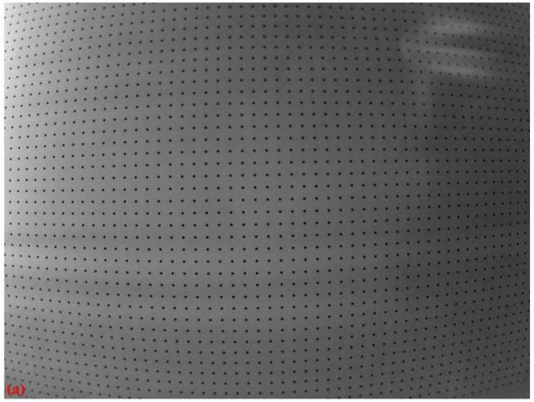
Method: https://discorpy.readthedocs.io/en/latest/tutorials/methods.html#correcting-perspective-effect

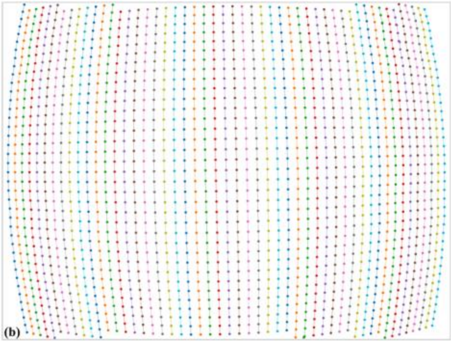
Usage: https://discorpy.readthedocs.io/en/latest/usage/demo_05.html

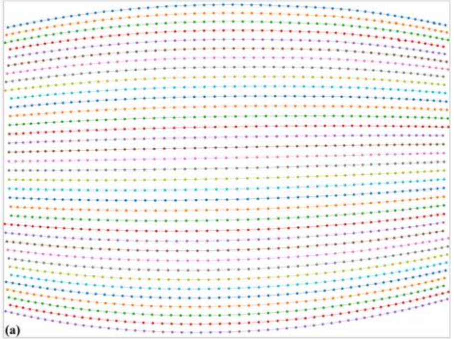
- Investigating, I found the Discorpy library (used by Migdal group and Australian CYGNUS)

- From an image with orthogonal dots, the barrel distortion and the perspective one can be recovered

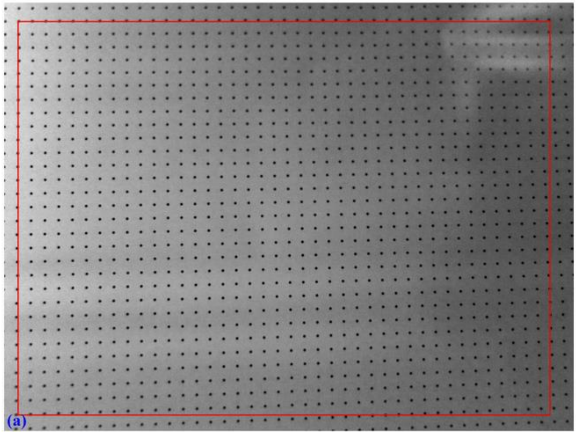
Image







Corrected



Barrel Correction

- Tested with a millimetric sheet. The code runs but image quality matters

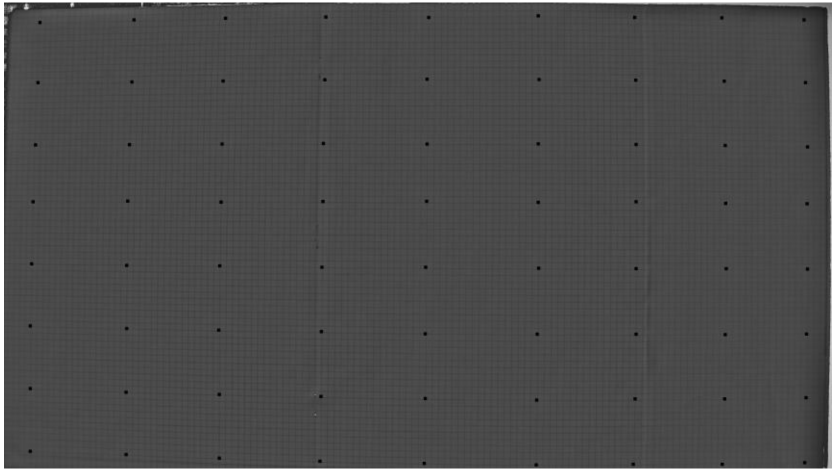
- The output is the list of the parameters of the function to correct position of pixels. Easy to implement (?)

- **Establishment of new methodology:** Every new optical system (lens, aperture, distance to object) needs to image and save a picture of a dotted surface on a grid nicely illuminated to have the barrel and perspective correction computed (for CYGNO-04, this will need to be done in advance of the data taking)

- A checkboard should work as well

- Dotted images can be manually created with webplodigitizer

- For perspective correction it is fundamental the same alignment is used as when data will be taken



It fails on this image:
- poor quality
- too large perspective misalignment?
- Too few lines?

Tests needed

After Reco?

- With the reco_run files some cluster variables are calculated, but it is not a full analysis
- For CYGNO-04, a different type of files will be generated by DAQ and the cluster analysis will be separated
- A new repository with postprocessing code is needed
- The idea is to provide the user a template to read the files and a group of functions the user can add to their custom code

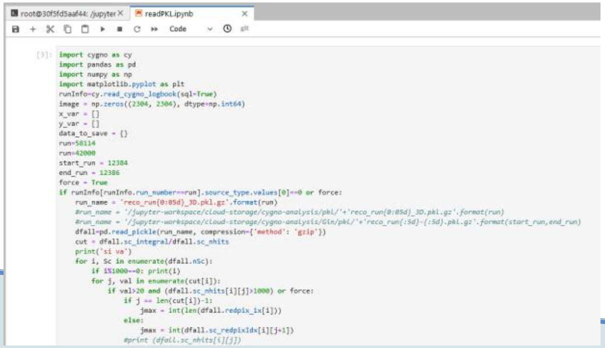
After Reco

Usage: https://root.cern/doc/v628/classROOT_1_1RDataFrame.html

- Type of code will depend on user preferences and purpose:

Python

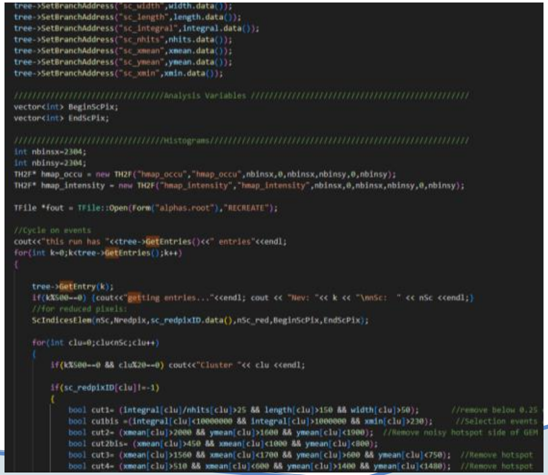
- Jupyter notebooks are excellent tools to debug, test pieces of code and produce specific plots
- With pandas dataframe with size smaller than the RAM they are very fast
- Pkl files created by monitor contain all information and are suited for python use
- Long and complex analysis will be slow



```python
import cygno as cy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
runInfo=cy.read_cygno_logbook(sql=True)
image = np.zeros((2304, 2304), dtype=np.int64)
x_var = []
y_var = []
data_to_save = {}
run=58114
run=42000
start_run = 12384
end_run = 12386
force = True
if runInfo[runInfo.run_number==run].source_type.values[0]==0 or force:
    run_name = 'reco_run{0:05d}_3D.pkl.gz'.format(run)
    #run_name = '/jupyter-workspace/cloud-storage/cygno-analysis/pkl/'+'reco_run{0:05d}_3D.pkl.gz'.format(run)
    #run_name = '/jupyter-workspace/cloud-storage/cygno-analysis/Gin/pkl/'+'reco_run{:5d}-{:5d}.pkl.gz'.format(start_run,end_run)
    dfall=pd.read_pickle(run_name, compression={'method': 'gzip'})
    cut = dfall.sc_integral/dfall.sc_nhits
    print('si va')
    for i, Sc in enumerate(dfall.nSc):
        if i%1000==0: print(i)
        for j, val in enumerate(cut[i]):
            if val>20 and (dfall.sc_nhits[i][j]>1000) or force:
                if j == len(cut[i])-1:
                    jmax = int(len(dfall.redpix_ix[i]))
                else:
                    jmax = int(dfall.sc_redpixIdx[i][j+1])
                #print (dfall.sc_nhits[i][j])
```

C++

- Online analysis can be performed with ROOT interface, but it is clunky and harder
- RDataFrame from ROOT and compilation of C++ should allow big data analysis the fastest and option
- Directionality is an example



```cpp
tree->SetBranchAddress("sc_width",width.data());
tree->SetBranchAddress("sc_length",length.data());
tree->SetBranchAddress("sc_integral",integral.data());
tree->SetBranchAddress("sc_nhits",nhits.data());
tree->SetBranchAddress("sc_xmean",xmean.data());
tree->SetBranchAddress("sc_ymean",ymean.data());
tree->SetBranchAddress("sc_xmin",xmin.data());

/////////////////////////////////////Analysis Variables ////////////////////////////////////////////////
vector<int> BeginScPix;
vector<int> EndScPix;

/////////////////////////////////////Histograms//////////////////////////////////////////////////////////
int nbinsx=2304;
int nbinsy=2304;
TH2F* hmap_occu = new TH2F("hmap_occu","hmap_occu",nbinsx,0,nbinsx,nbinsy,0,nbinsy);
TH2F* hmap_intensity = new TH2F("hmap_intensity","hmap_intensity",nbinsx,0,nbinsx,nbinsy,0,nbinsy);

TFile *fout = TFile::Open(Form("alphas.root"),"RECREATE");

//Cycle on events
cout<<"this run has "<<tree->GetEntries()<<" entries"<<endl;
for(int k=0;k<tree->GetEntries();k++)
{

   tree->GetEntry(k);
   if(k%500==0) {cout<<"getting entries..."<<endl; cout << "Nev: "<< k << "\nnSc:  " << nSc <<endl;}
   //for reduced pixels:
   ScIndicesElem(nSc,Nredpix,sc_redpixID.data(),nSc_red,BeginScPix,EndScPix);

   for(int clu=0;clu<nSc;clu++)
   {
      if(k%500==0 && clu%20==0) cout<<"Cluster "<< clu <<endl;

      if(sc_redpixID[clu]!=-1)
      {
         bool cut1= (integral[clu]/nhits[clu]>25 && length[clu]>150 && width[clu]>50);       //remove below 0.25
         bool cu1bis =(integral[clu]<10000000 && integral[clu]>1000000 && xmin[clu]>230);     //Selection events
         bool cut2= (xmean[clu]>2000 && ymean[clu]>1600 && ymean[clu]<1900);  //Remove noisy hotspot side of GEM
         bool cut2bis= (xmean[clu]>450 && xmean[clu]<1000 && ymean[clu]<800);
         bool cut3= (xmean[clu]>1560 && xmean[clu]<1700 && ymean[clu]>600 && ymean[clu]<750);  //Remove hotspot
         bool cut4= (xmean[clu]>510 && xmean[clu]<600 && ymean[clu]>1400 && ymean[clu]<1480);  //Remove hotspot
```

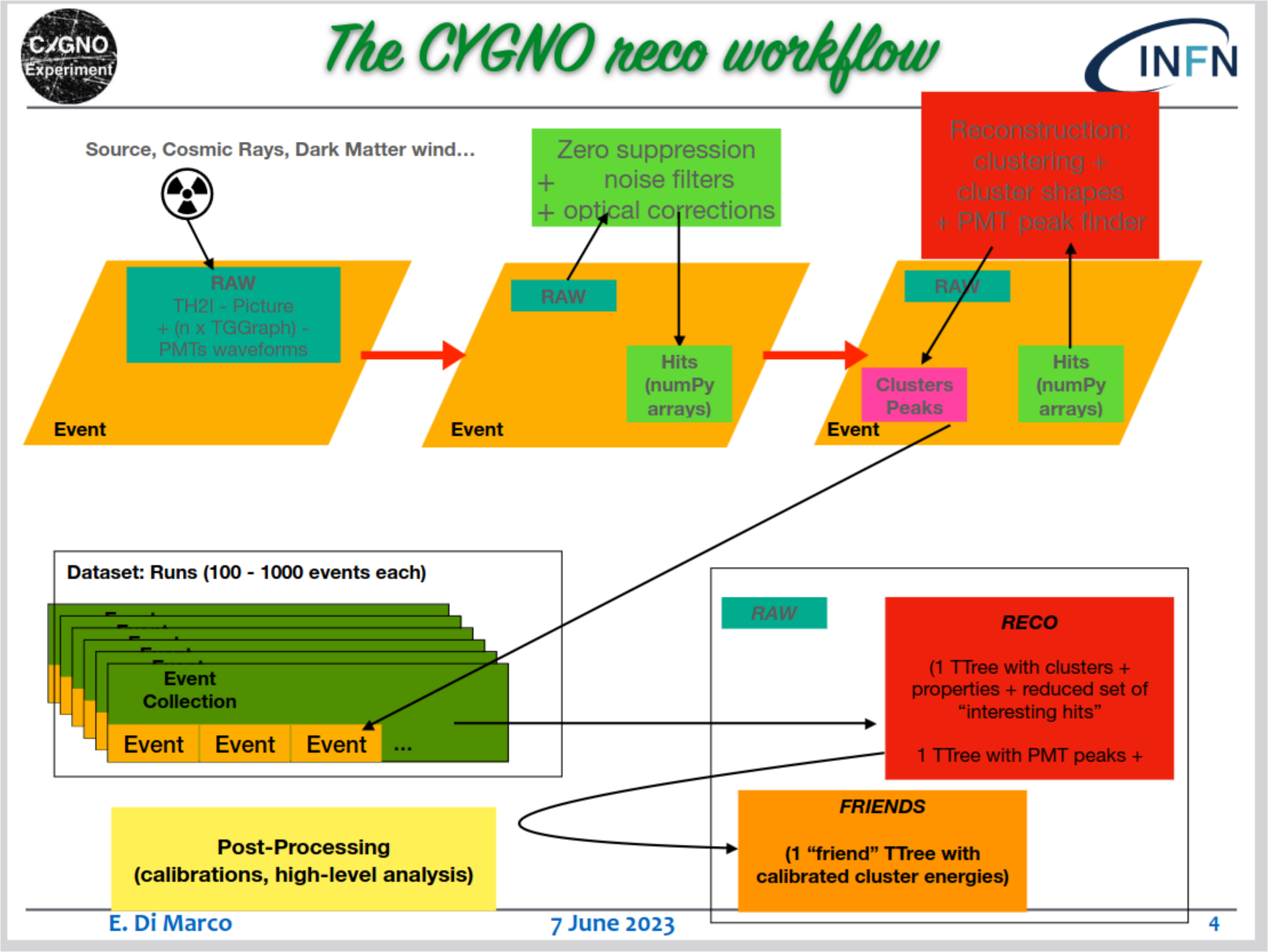
Conclusions

- Big improvements were achieved this year with focus on stability
- Additions of PMT, QUEST and rotation fix are completed in compliance with CYGNO-04 approaching
- Vignette map code ready to provide correction map for all prototypes
- Barreling and perspective correction under study

Future

- Disuniformity map could also be added in future once understood
- Focus will shift towards
    - coping with new DAQ data format
    - Higher level analysis

Extra 1



The CYGNO reco workflow

CYGNO Experiment

INFN

Source, Cosmic Rays, Dark Matter wind…

RAW
TH2I - Picture
+ (n x TGGraph) -
PMTs waveforms

Event

Zero suppression
+ noise filters
+ optical corrections

RAW

Hits
(numPy
arrays)

Event

Reconstruction:
clustering +
cluster shapes
+ PMT peak finder

RAW

Clusters
Peaks

Hits
(numPy
arrays)

Event

Dataset: Runs (100 - 1000 events each)

Event
Collection

Event Event Event …

RAW

RECO

(1 TTree with clusters + properties + reduced set of "interesting hits"

1 TTree with PMT peaks +

FRIENDS

(1 "friend" TTree with calibrated cluster energies)

Post-Processing
(calibrations, high-level analysis)

E. Di Marco 7 June 2023 4