FNG data

LogBook

May 17, 2019

FNG DATA

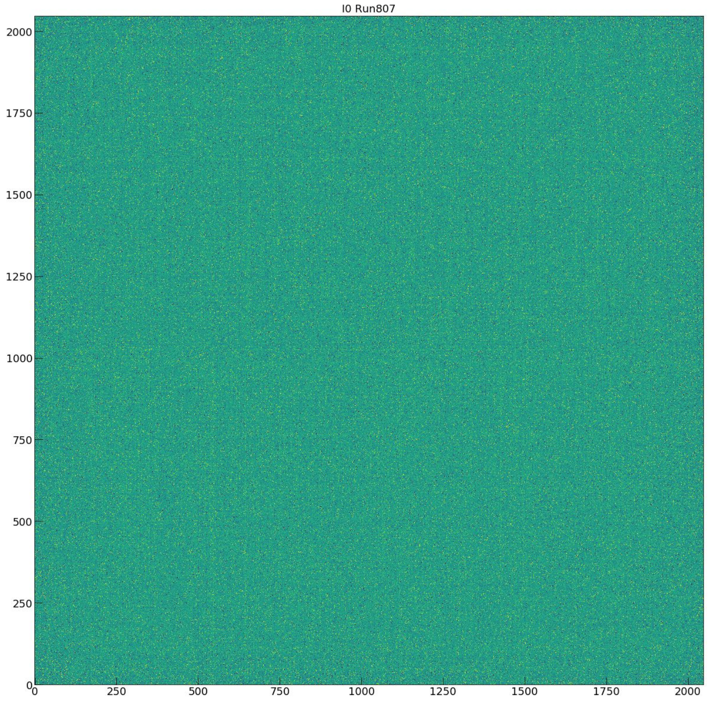
- Separate Nuclear Recoil from everything else;
- Evaluate Eff vs Length

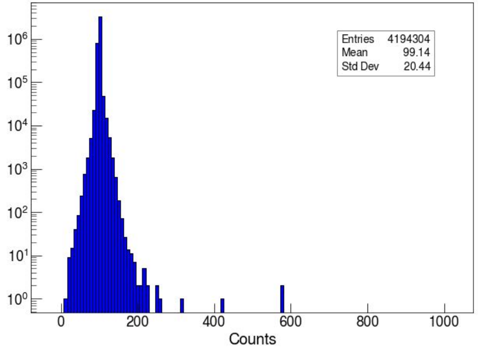
- Get the parameters:
    - Length
    - Width
    - Photons/pixel
    - Profile
    - Etc

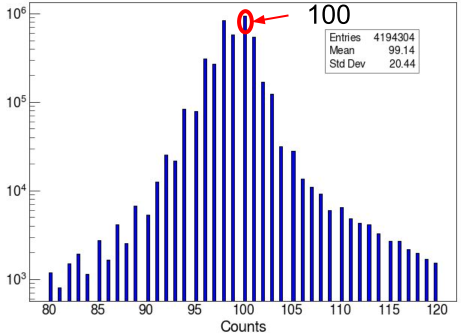
About the FNG data

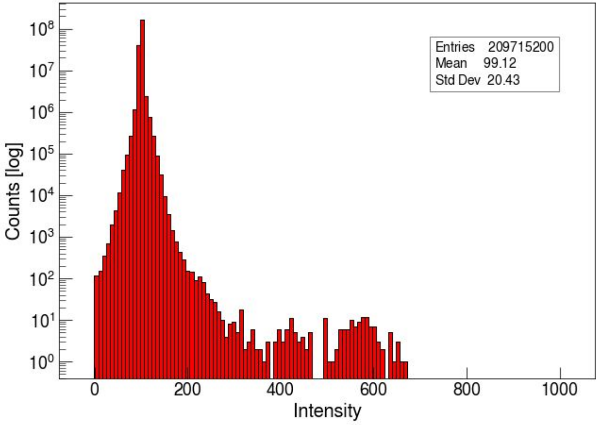
We have the Run 807 with BEAM OFF

GEM Voltage (V)	He:CF4	Transfer field strength (kV/cm)	CMOS Exposure Time (ms)	Nominal Flux cubic cm/min	Effective flux cubic cm/min	Acquisition Number (# Events)	Up Voltage (V) (fixed)
420	60/40 premix	2	100	300	218.4	100	2060





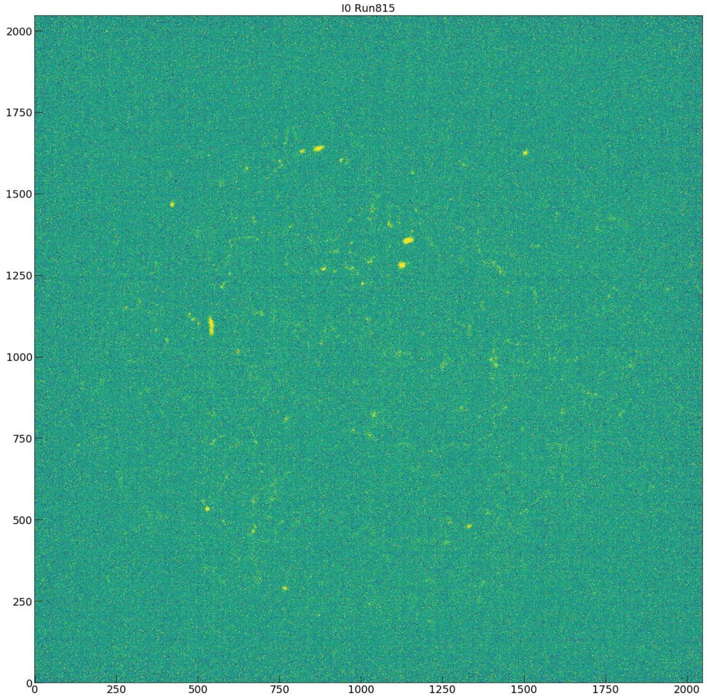


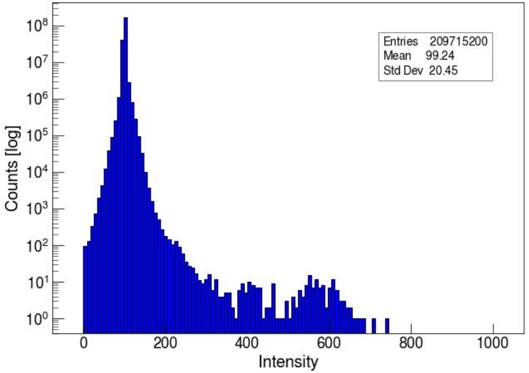


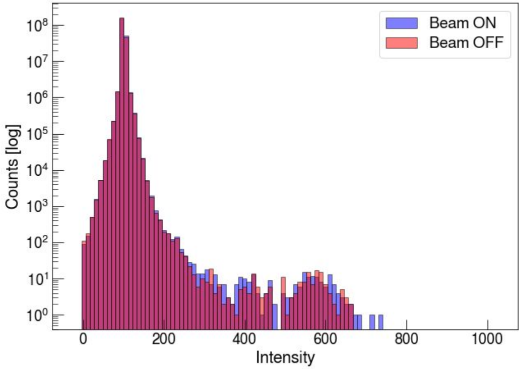
About the FNG data

We have the Run 815 with BEAM ON

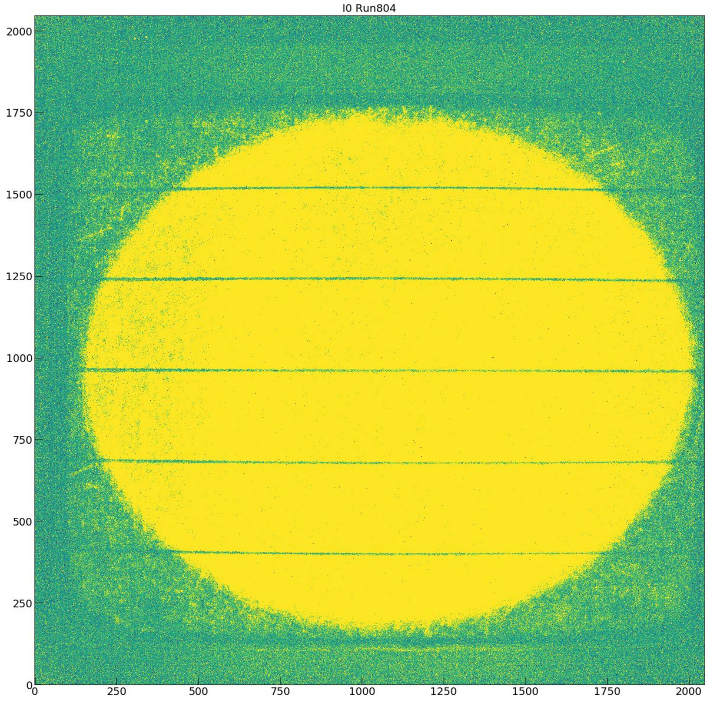
GEM Voltage (V)	He:CF4	Transfer field strength (kV/cm)	CMOS Exposure Time (ms)	Nominal Flux cubic cm/min	Effective flux cubic cm/min	Acquisition Number (# Events)	Up Voltage (V) (fixed)
440	60/40 premix	2	100	300	218.4	300	2120







About the FNG data



We have the Run 804 with BEAM ON

GEM Voltage (V)	He:CF4	Transfer field strength (kV/cm)	CMOS Exposure Time (ms)	Nominal Flux cubic cm/min	Effective flux cubic cm/min	Acquisition Number (# Events)	Up Voltage (V) (fixed)
420	60/40 premix	2	10000	300	218.4	10	2060

Here we can observe the ‘Sensitive’ region of the detector.

Could we simply exclude everything out of this ellipse?

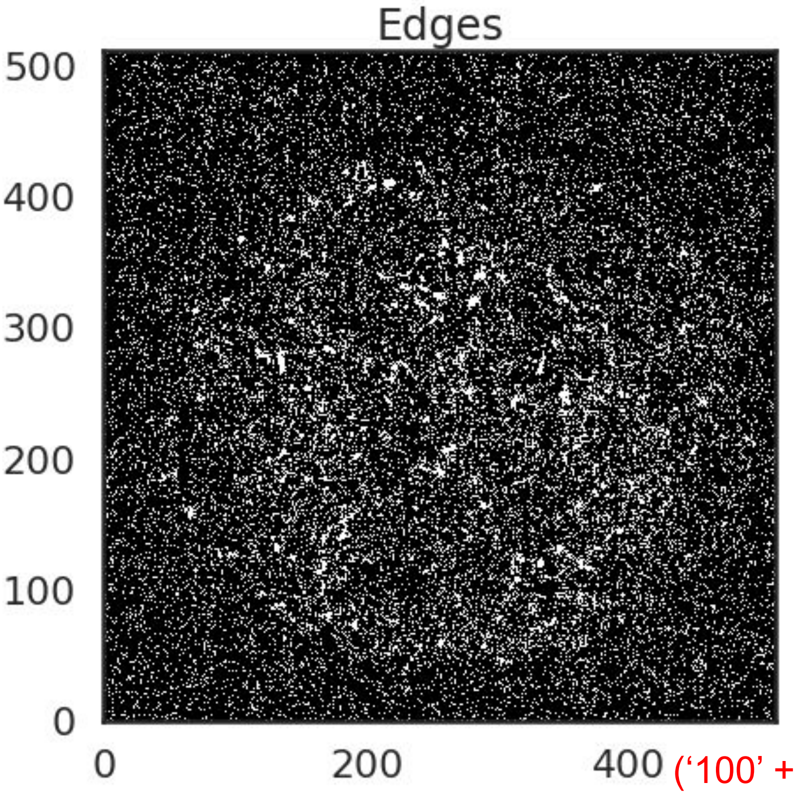
NO, it is better to stay analysing everything.

Subtracting Pedestal

Subtracting Pedestal

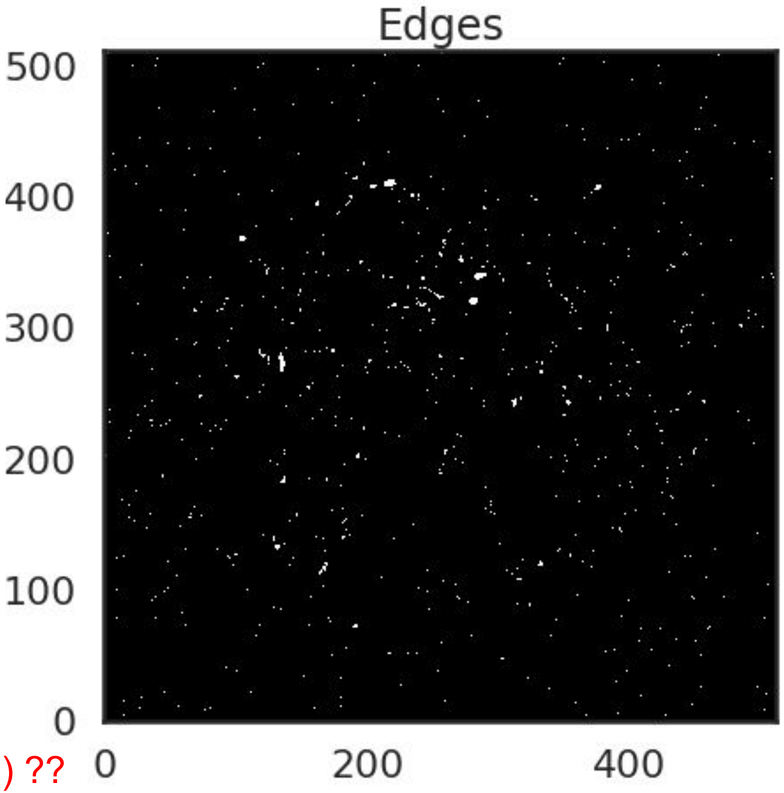
we are using the RUN817 to make the pedmap

Subtracting ‘100’



(‘100’ + sigma) ??

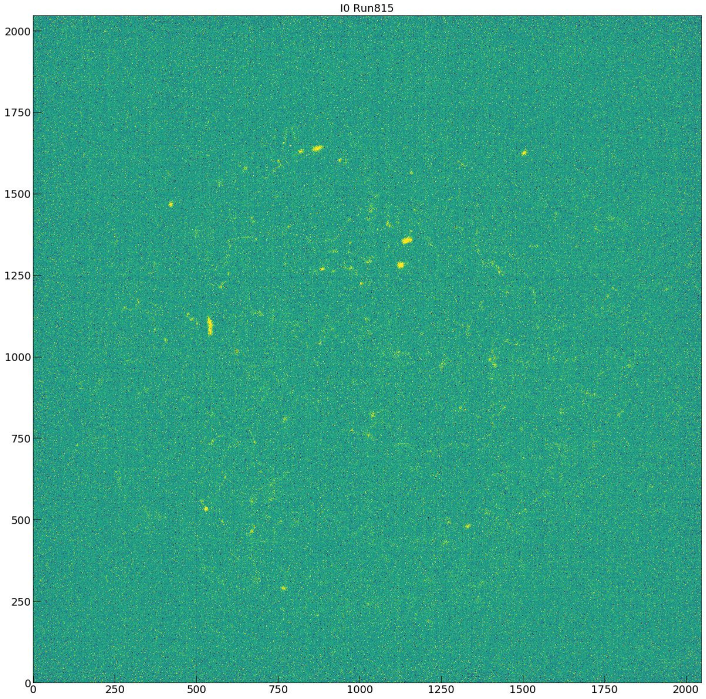
Using PEDMAP (mean + sigma)



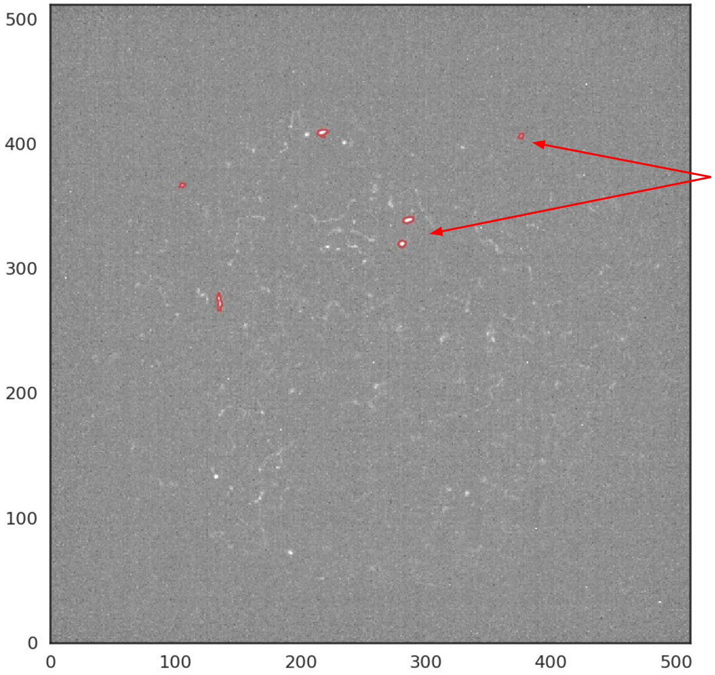
Setting i2DBSCAN

i2DBSCAN to find Nuclear Recoil

Original Image



Rescale Image

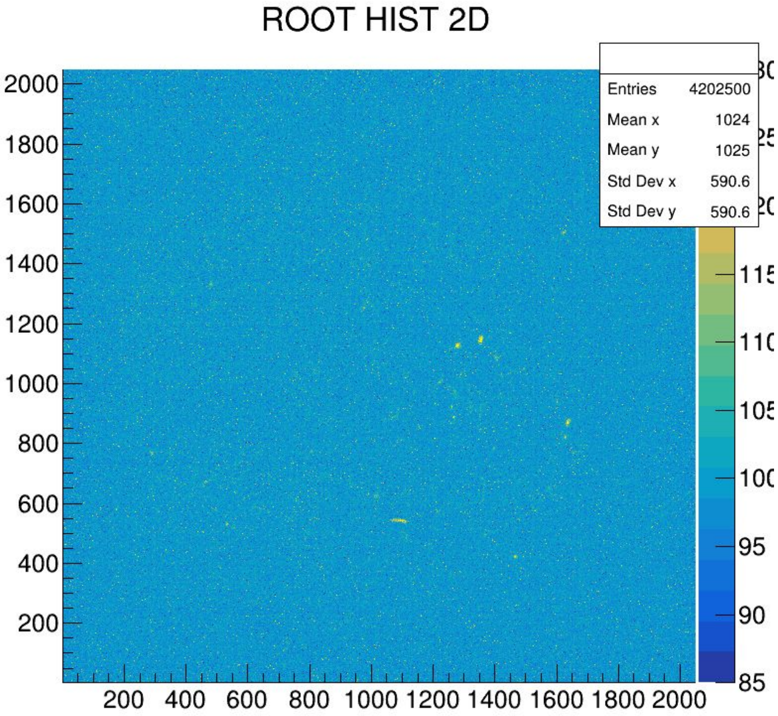


Found cluster

Tasks

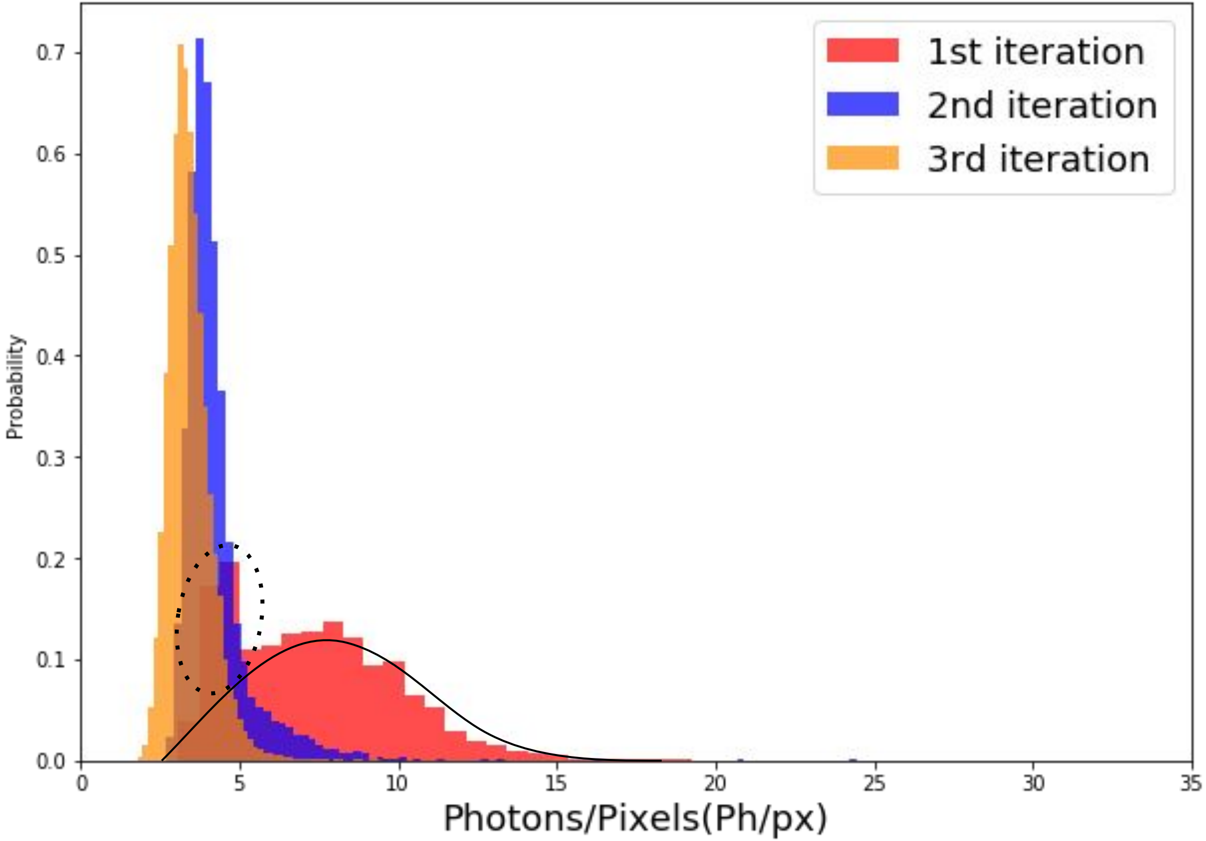
Implementation of the Algorithm

- Emanuele environment;
- I2DBSCAN attached;
- Py2 to Py3 converted;
- Changed array to ROOT hist;
- Changing the ‘Run loading method‘ to get the run on the cloud;
- Debugging the algorithm
- Adding new variables
- Algorithm to follow the worm (not started)



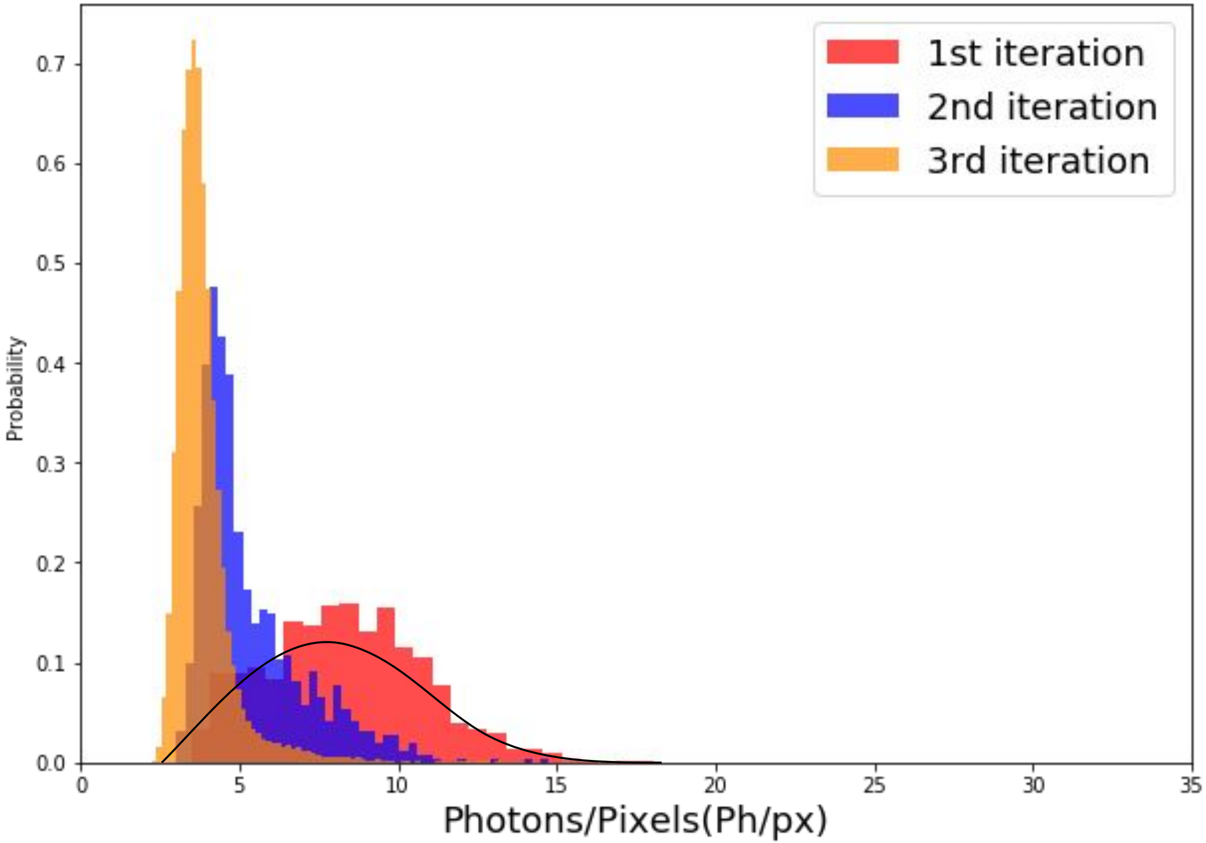
Python environment

Cluster reconstruction



- It seems that the 2nd and 3rd iterations are getting background.
- And in the left part of the 1st iteration we are having some ‘contamination’;
- Maybe will be better to set tight the parameters of the 1st and 2nd iterations;

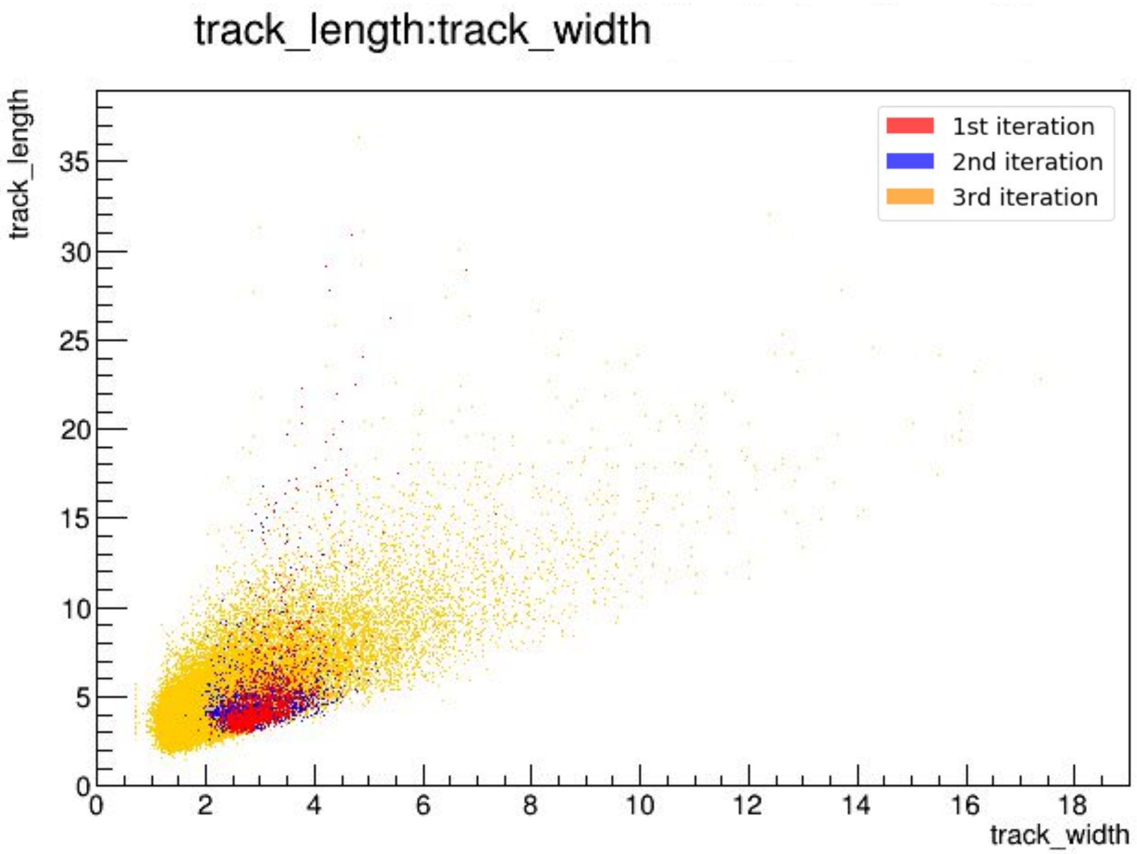
Cluster reconstruction



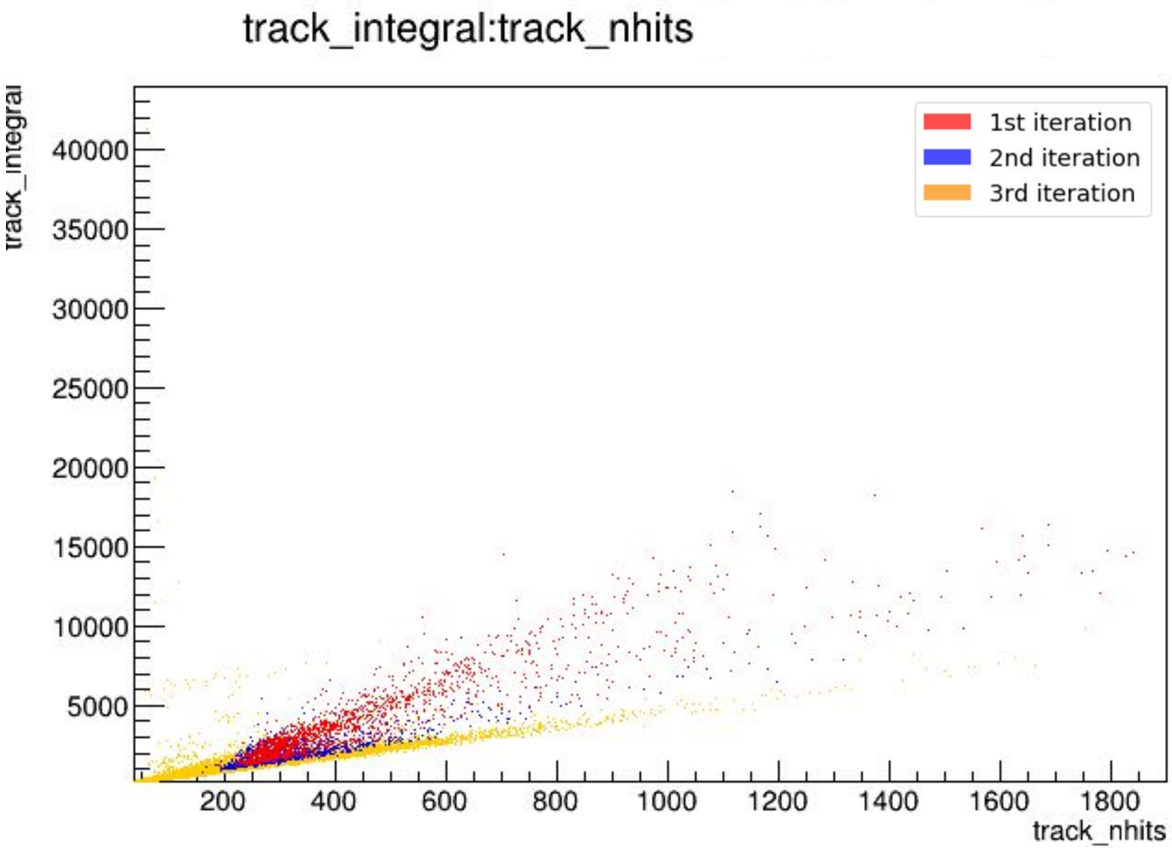
- Now, setting the tighter the parameters of the 1st and 2nd iterations;
- We can see less ‘contamination’ on the red distribution.

PyROOT environment

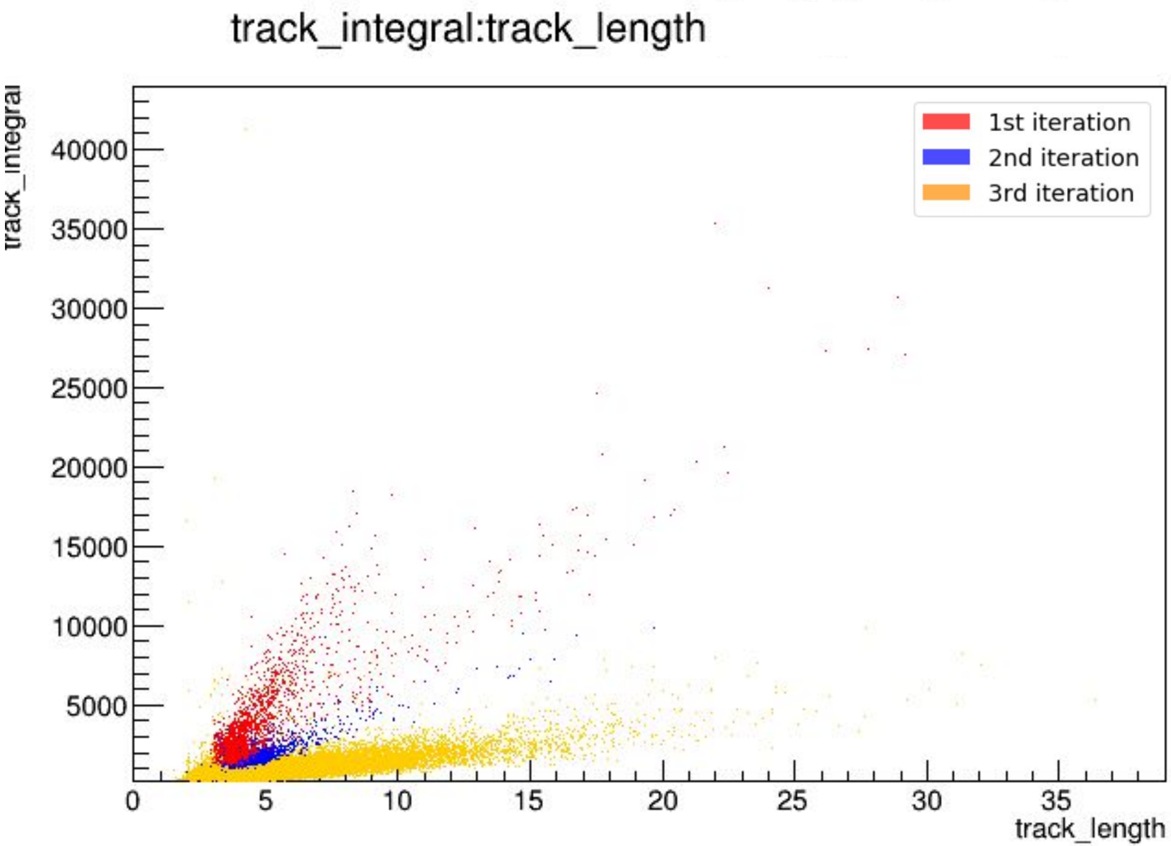
Three iterations divided into different colours



Three iterations divided into different colours



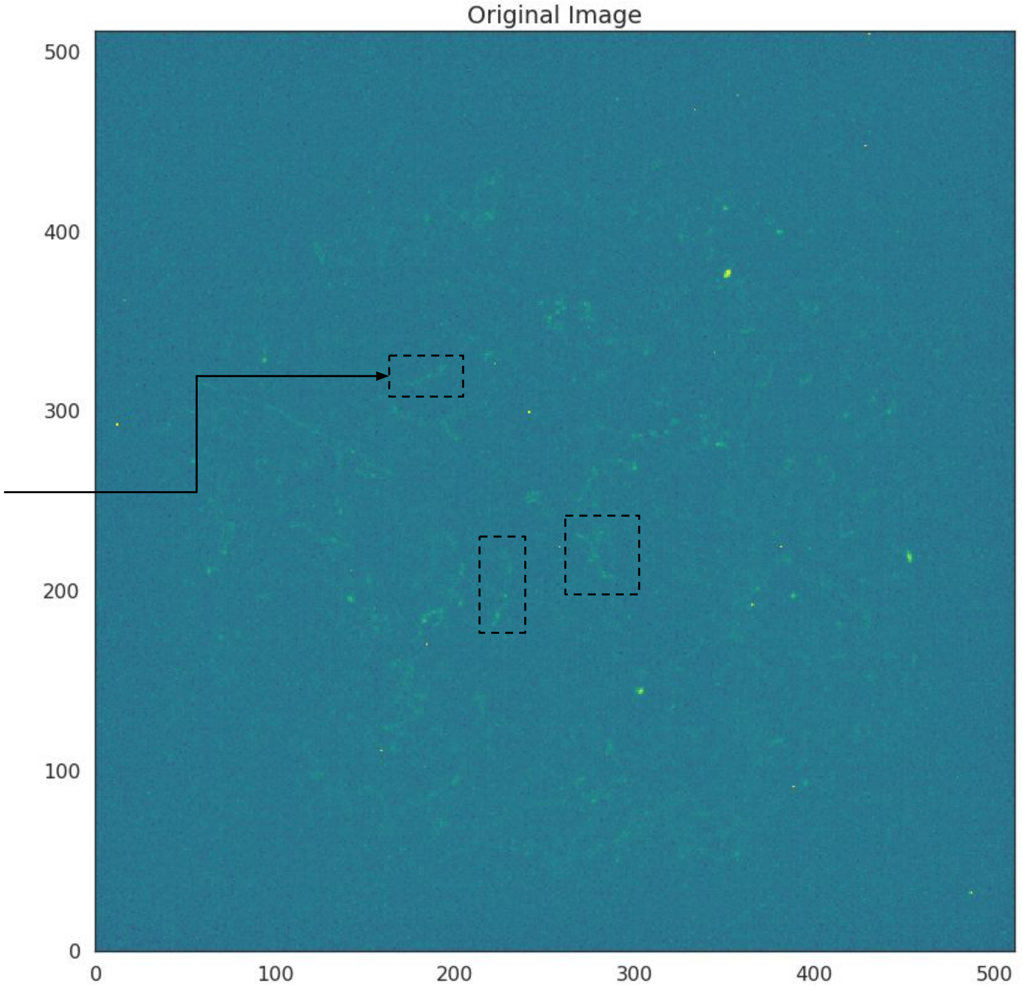
Three iterations divided into different colours



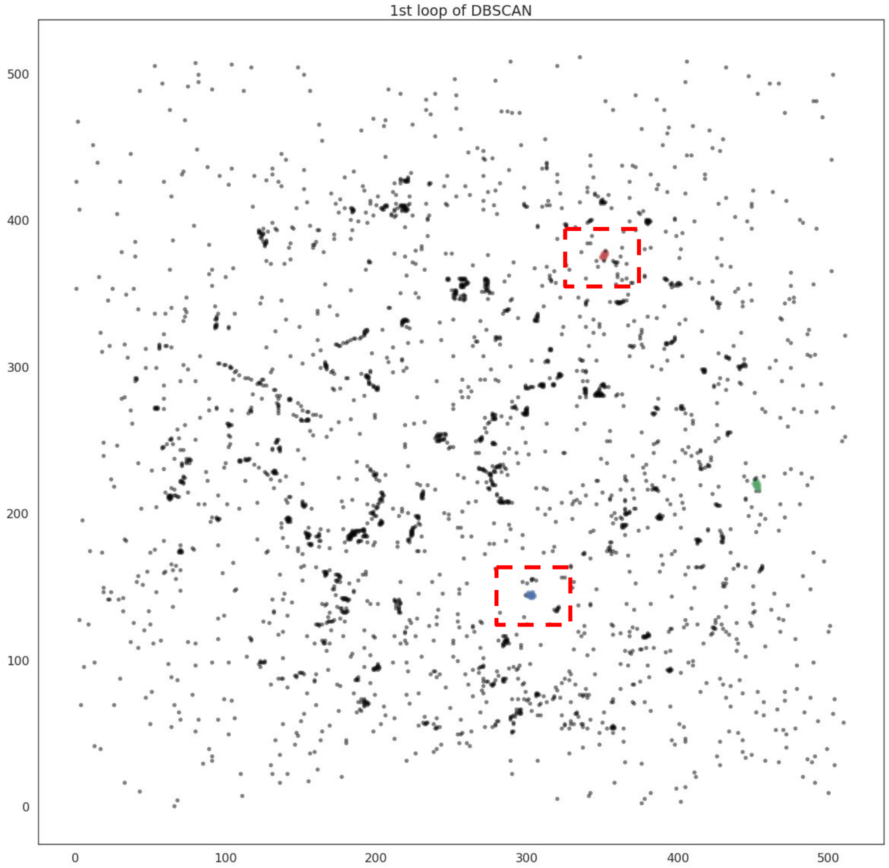
Run815 - Image 1

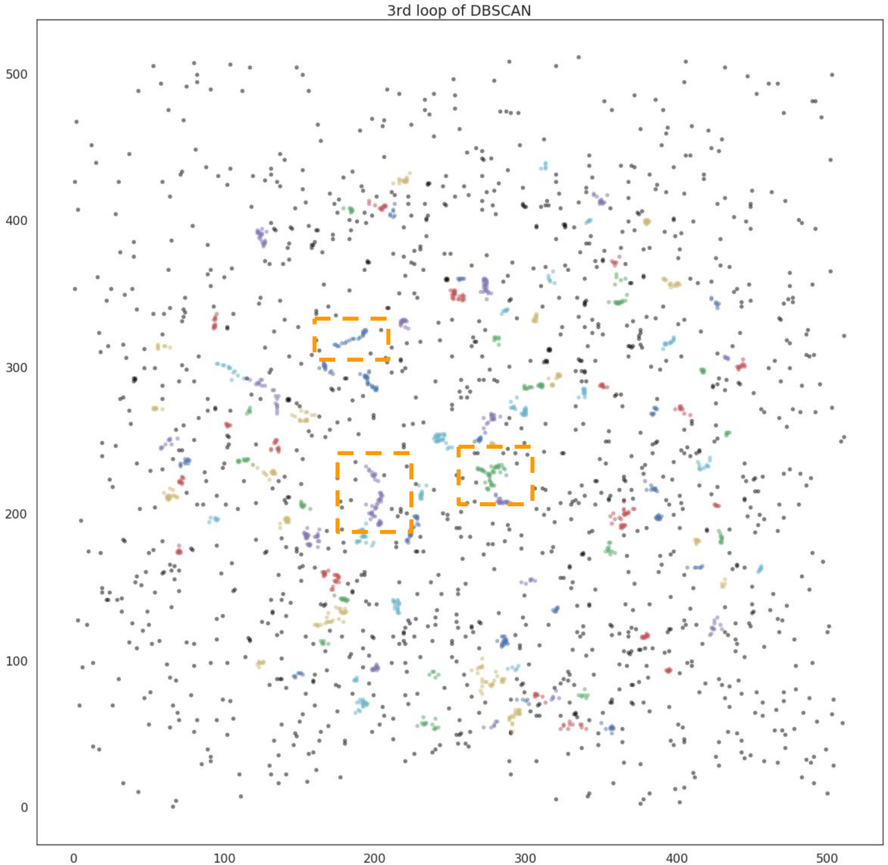
This is one kind of the clusters that are been getting on the 3rd iteration…

As you will see the clusters found on the 3rd iteration can have a length bigger than the ones from 1st iteration

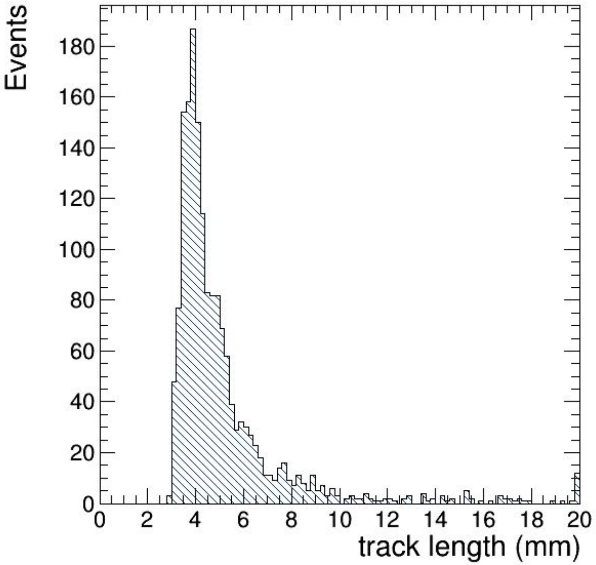


Run815 - Image 1 (different colour is different clusters)

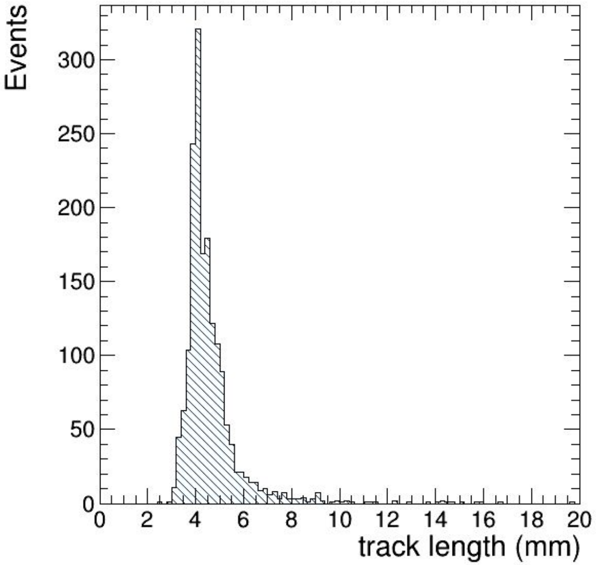

1st loop of DBSCAN


3rd loop of DBSCAN

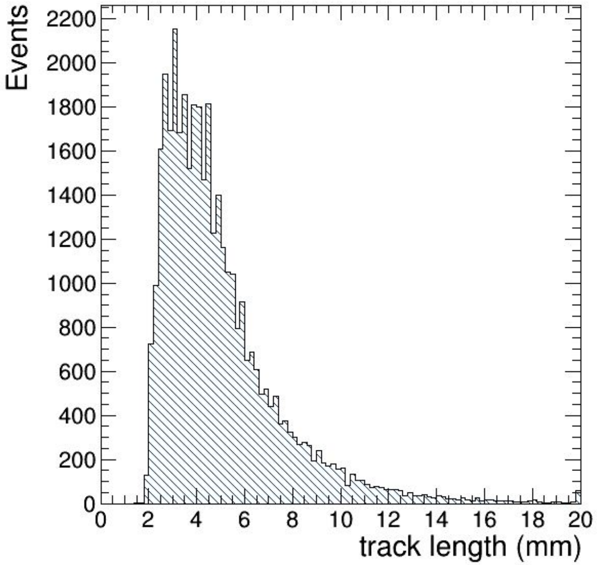
Cluster length in the cluster - Run 815



ITER 1

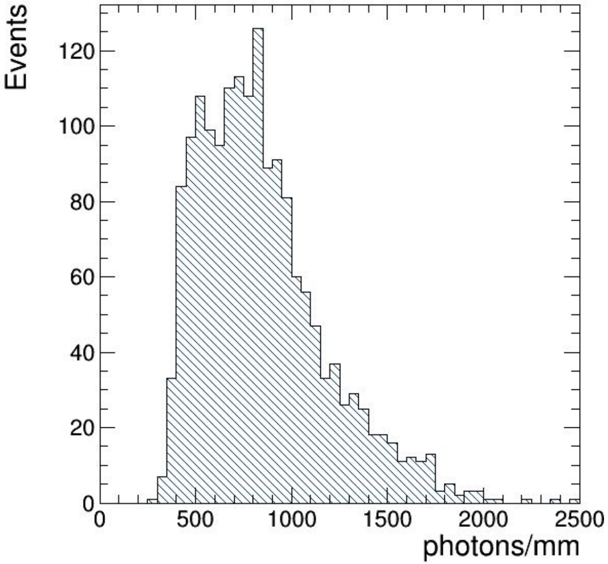


ITER 2

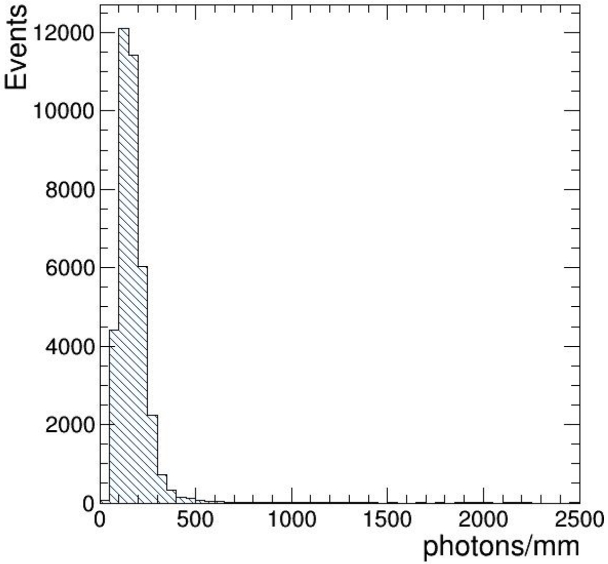


ITER 3

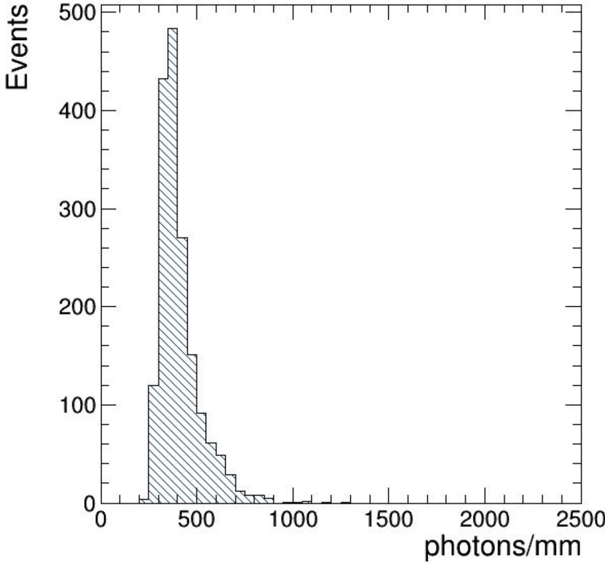
Number of photons / mm in the cluster - Run 815



ITER 1

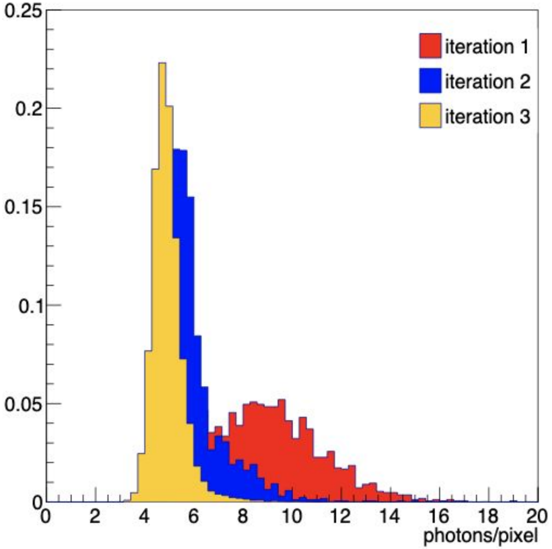


ITER 2

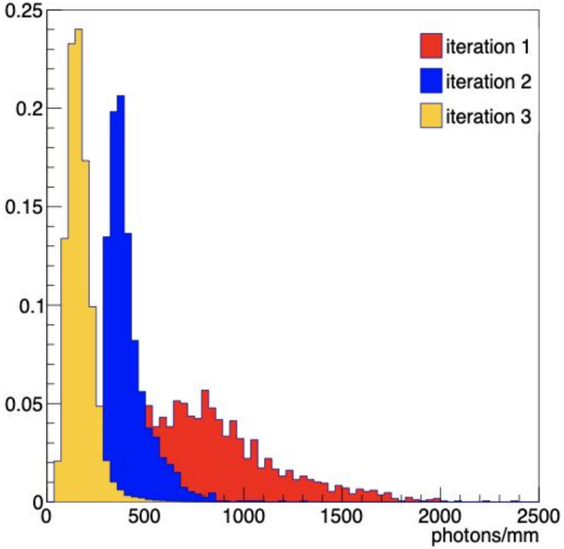


ITER 3

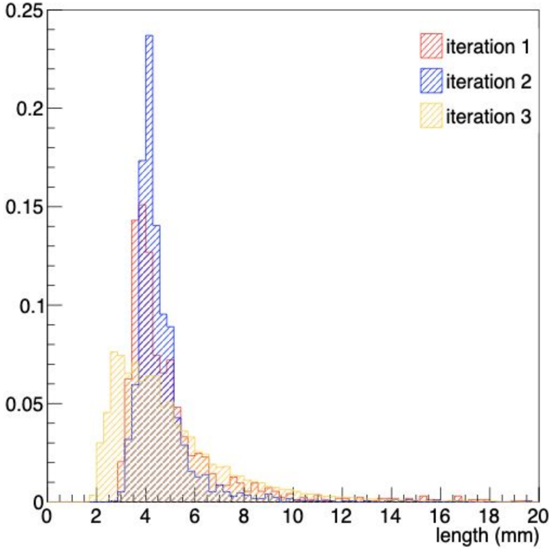
3 iterations on Beam ON run



ph/pixels



photons/mm (~dE/dx)



Cluster length (mm)

Studies to improve DBSCAN

Studies on how to improve DBSCAN

- DBSCAN 3D - Using the Z-axis (counts):
    - DBSCAN works in 3D and I have test no FNG data.
    - My first impression is that using 3D on DBSCAN will not lead us to an improve, because when we look at the 3 dimensions the **sparsity** on the 'nuclear recoils' is greater than in the others, because of the 'amplitude'.
- Automatization of the *eps, minpoints* tuning:
    - I have started to study the possibility of it. For now I found that it isn't so simple, but I think is doable. Of course that will not be better than manual tuning, however it is important for future of the clustering algorithm. Maybe something smart and **supervised**.

Studies on how to improve DBSCAN

- DBSCAN “3D” - Using the Z-axis (counts):
  - As DBSCAN uses proximity (eps) and minimum number of points inside some area (minPoints) to find the clusters, we thought to use the Z-axis on the 2D dimension;
  - Until now, if a pixel passes on the threshold, it will be an input to DBSCAN as a single coordinate, not taking in account the number of ‘photons’ in that pixel.
  - **The idea here is**: get the coordinate of the pixel (x,y) that passed throught the threshold and replicate this coordinate by the number of ‘photons’.
  - In this way will be possible to ‘simulate’ the third dimension without have to lead of the ‘curse of dimensionality’;
  - And in theory, we will be able to ‘tight’ more the *MinPoint* parameters in order to be more efficiency in getting the ‘Nuclear Recoils’.