Cross-talk and sigma studies in GaAs Simulation

Semen Doronin
Nikita Belyaev

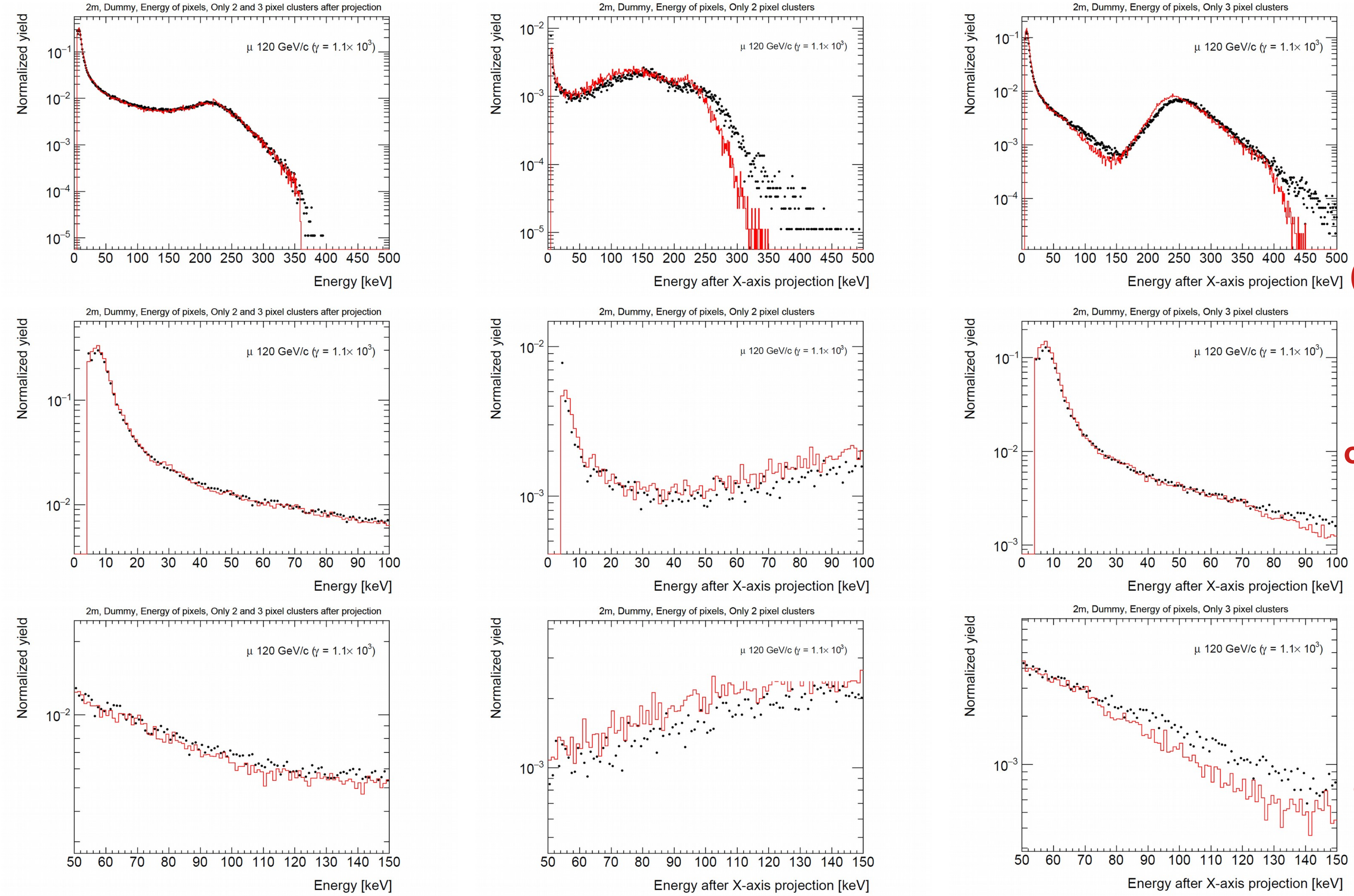
Study of the main distributions with per pixels energy calibration

Parameters of the simulation model:

1) 2m distance between radiator and GaAs-detector, Dummy radiator
2) 120GeV muon beam (7 run — 2018)
3) 0.7 degree rotation of the GaAs-detector
4) Sides+corners clusterization
5) Energy deposition of the particle in each slice based on Landau distribution with $E_{mpv}/10$ and sigma = 1.8
6) $E_{threshold}$ = 3.7keV, σ_{noize} = 0.426keV, σ_{ToT} = 0.8keV, D = 110 cm^2/s, V_{drift} = 1.05*10^7 cm/s, E_{mpv} = 260keV
7) Different capacity matrix
8) Fixed particle diffusion:

$$\sigma_0 = 0.008\,E^{1.7} + 0.4,\ \sigma = \sqrt{2 * D * (500 - z)/V_{drift}},\ \sigma_{total} = \sqrt{\sigma_0^2 + \sigma^2}$$

Capacity matrix (-19% from initial):(0.02025, 0.00081, 0.02025, 0.00081, 0.02025, 0.00081, 0.02025, 0.00081)



7 run (2018)

After one step calibration (New, Seventh)

<u>Version 2</u>:

Sqrt, $\sigma_0=0.008E^{1.7}+0.4$

D = 110 cm²/s

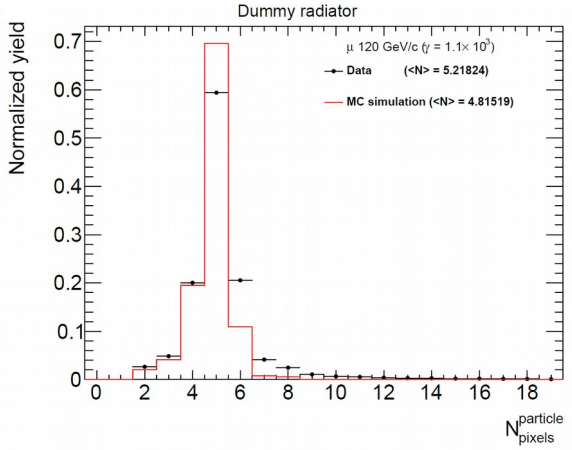
0.7 degree

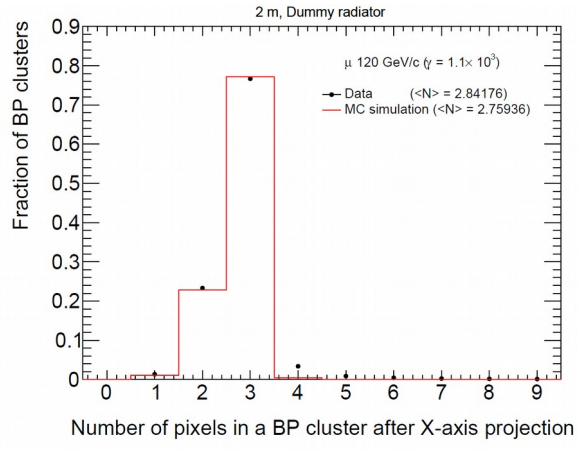
E_{thr} = 3.7keV

Normalization on the number of particles

Capacity matrix (-19% from initial):(0.02025, 0.00081, 0.02025, 0.00081, 0.02025, 0.00081, 0.02025, 0.00081)

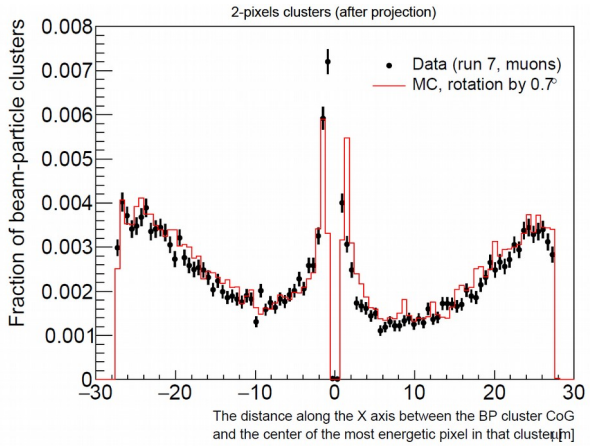
X-axis projection

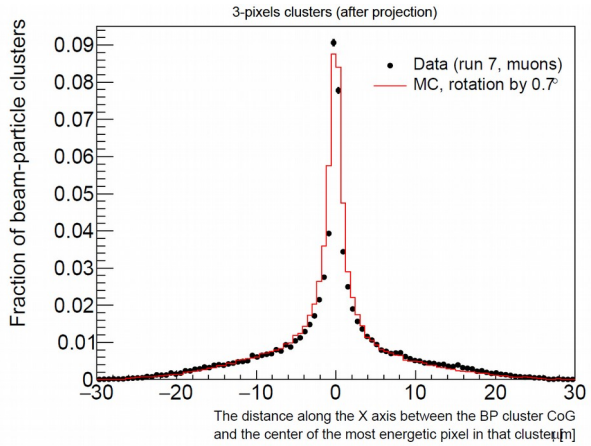


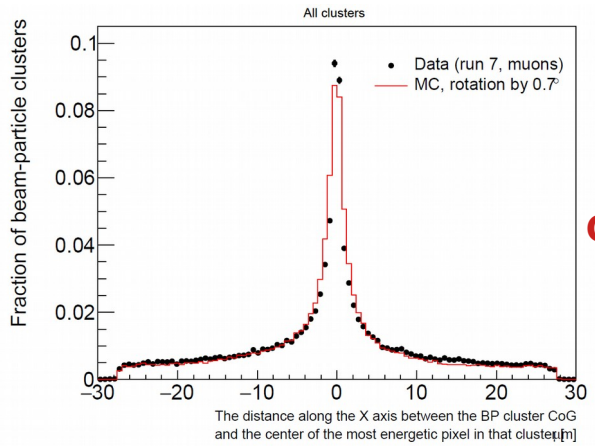


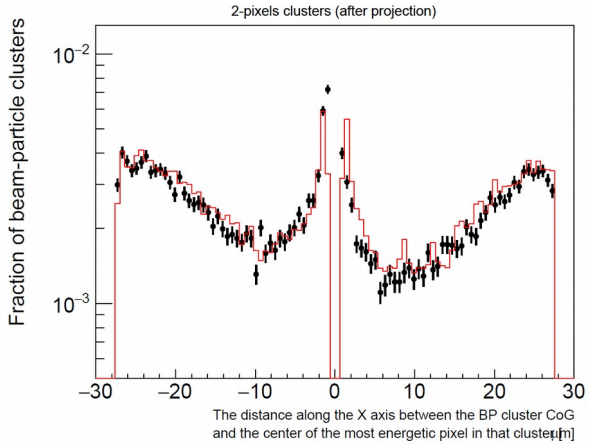
7 run (2018)

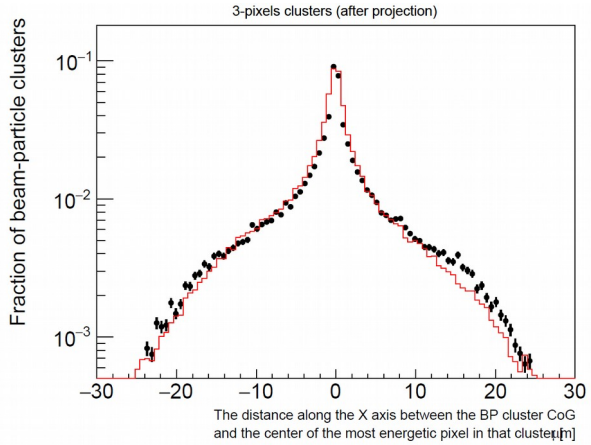
After one step calibration (New, Seventh)

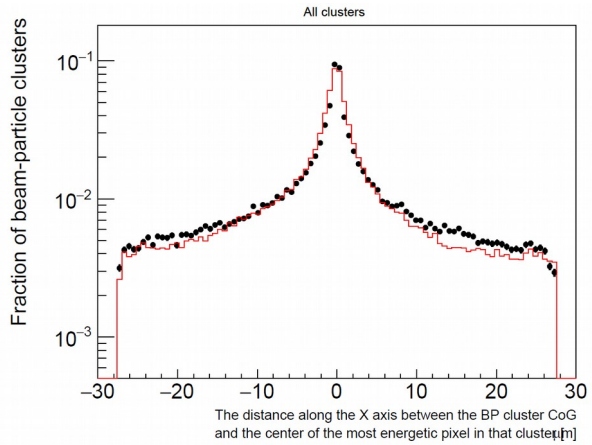












Version 2:

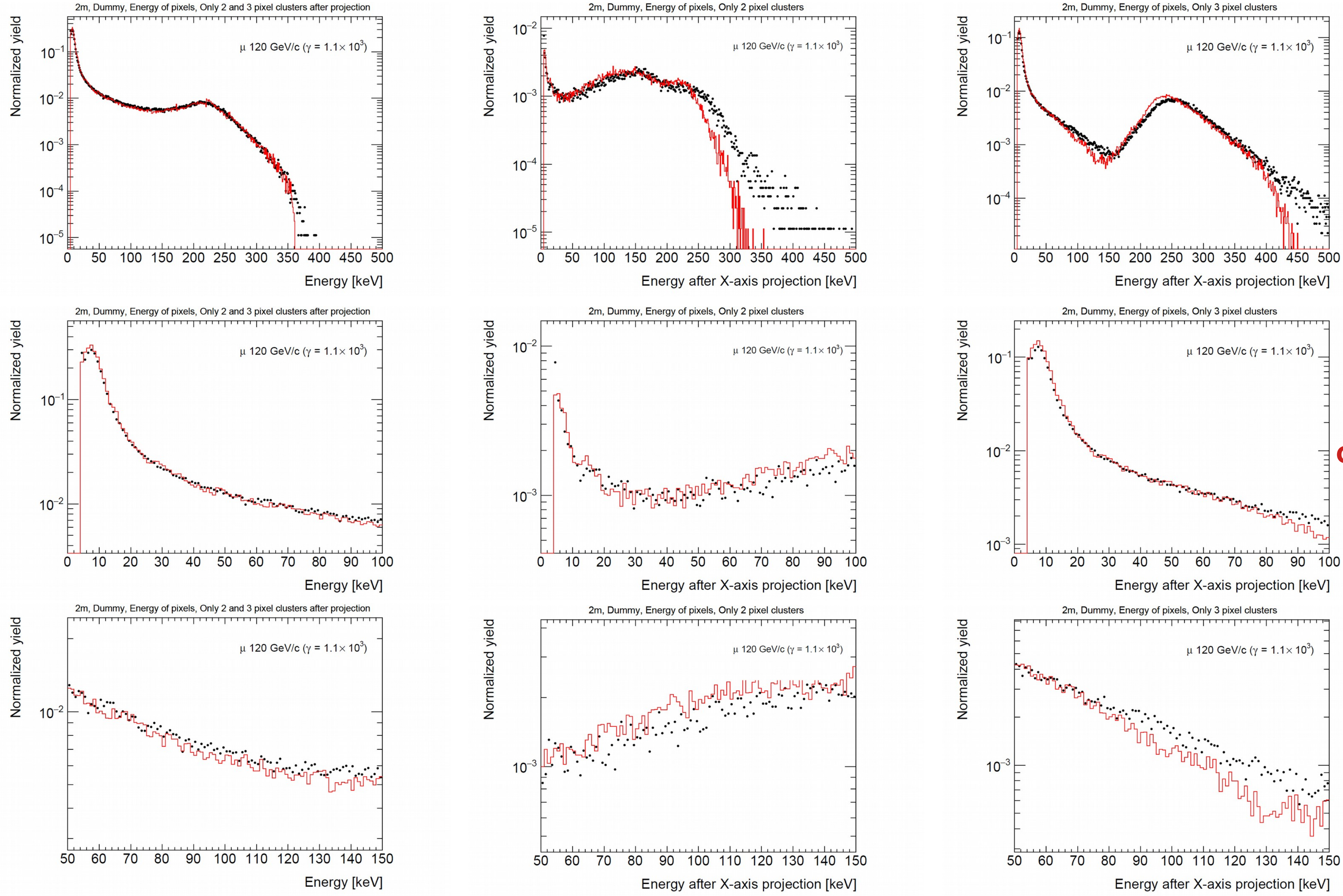
Sqrt, $\sigma_0 = 0.008E^{1.7} + 0.4$

D = 110 cm²/s

0.7 degree

E_{thr} = 3.7keV

Capacity matrix (-18% from initial):(0.0205, 0.00082, 0.0205, 0.00082, 0.0205, 0.00082, 0.0205, 0.00082)



7 run (2018)

After one step calibration (New, Seventh)

Version 2:

Sqrt, σ_0=0.008E$^{1.7}$+0.4

D = 110 cm²/s

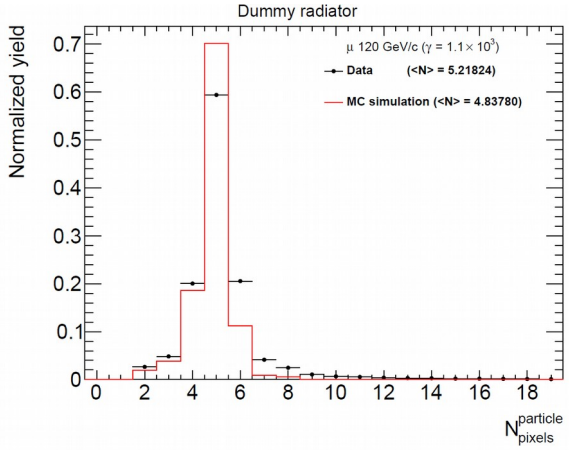
0.7 degree

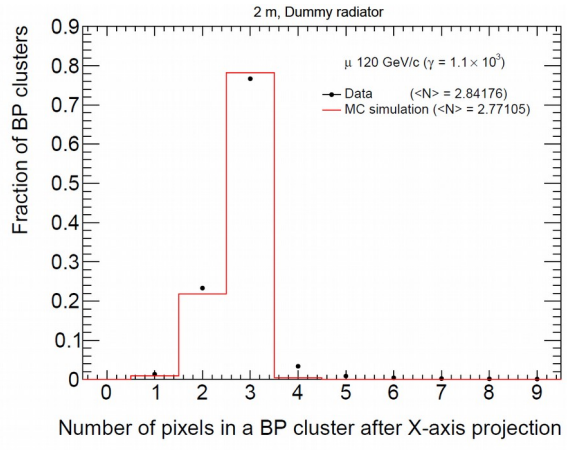
E_{thr} = 3.7keV

Normalization on the number of particles

Capacity matrix (-18% from initial):(0.0205, 0.00082, 0.0205, 0.00082, 0.0205, 0.00082, 0.0205, 0.00082)

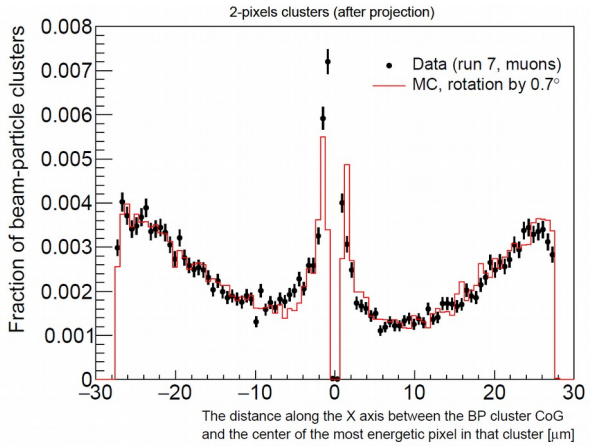
X-axis projection

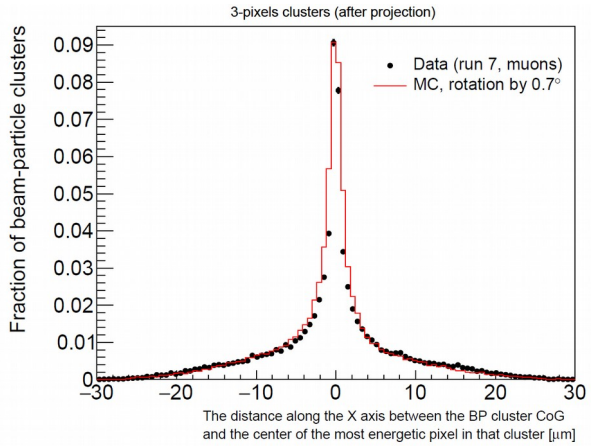


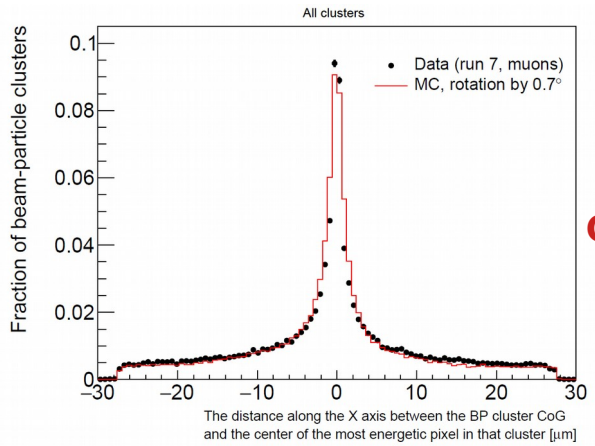


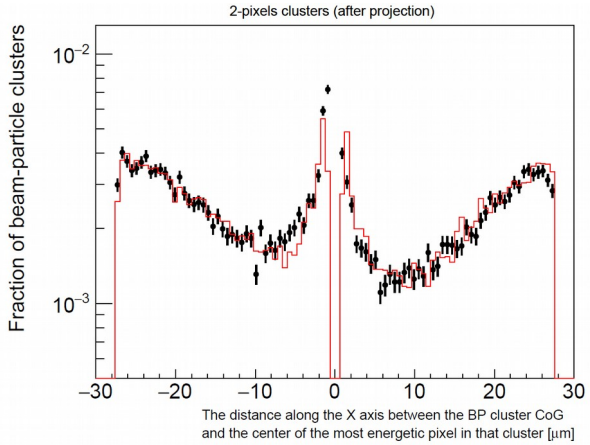
7 run (2018)

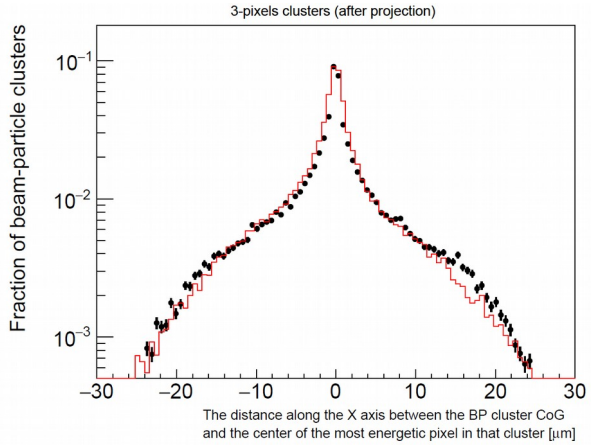
After one step calibration (New, Seventh)

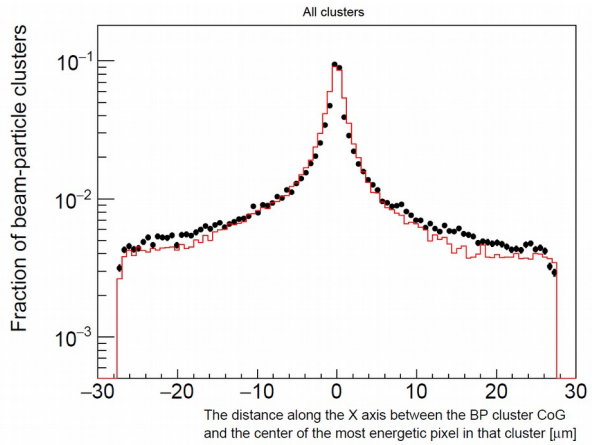












Version 2:

Sqrt, σ_0=0.008E$^{1.7}$+0.4

D = 110 cm^2/s

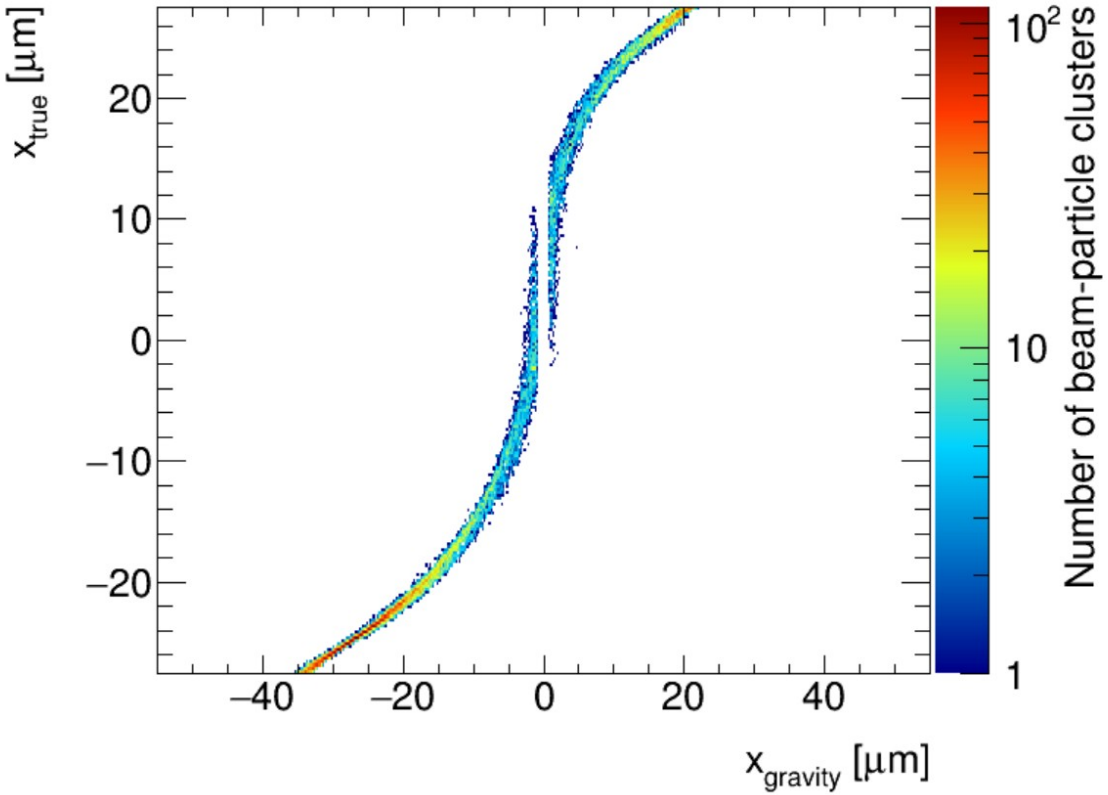
0.7 degree

E_{thr} = 3.7keV

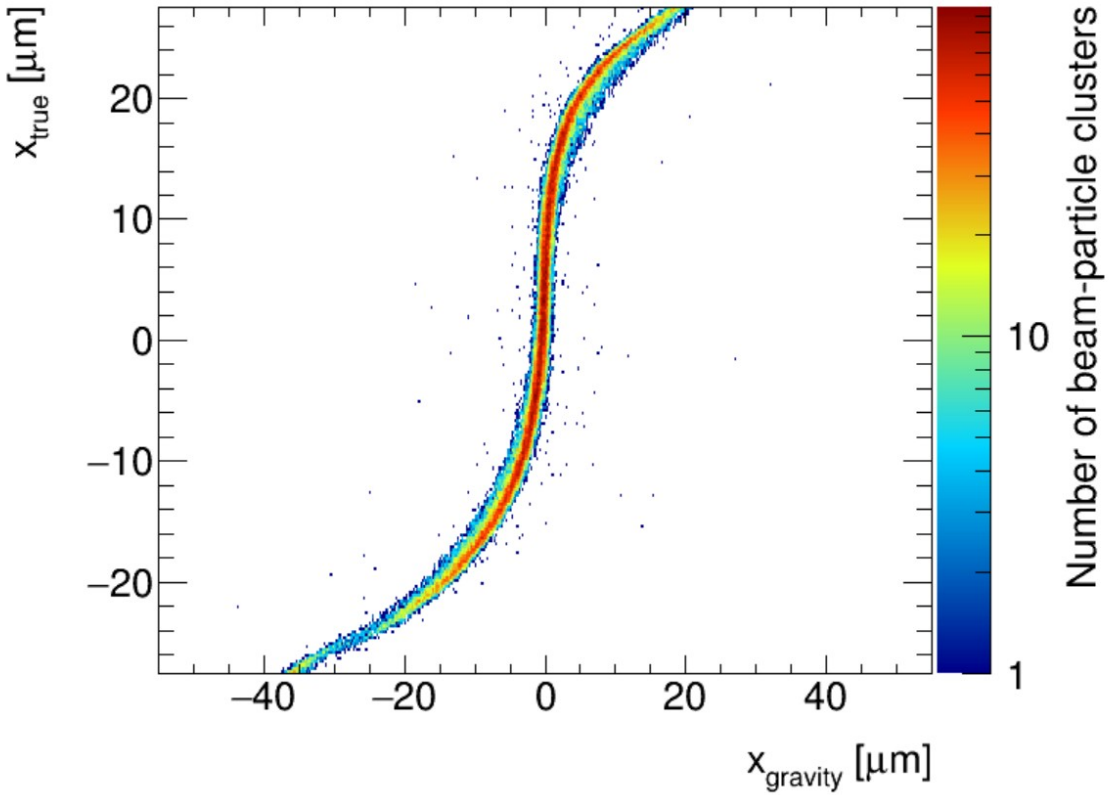
S-curves from ~sdoronin/public/Simulation_outfile_EnergyCalibration.root

Muons, dummy radiator, sides+corners

Clusters consisting of exactly 2 pixels after X-axis projection



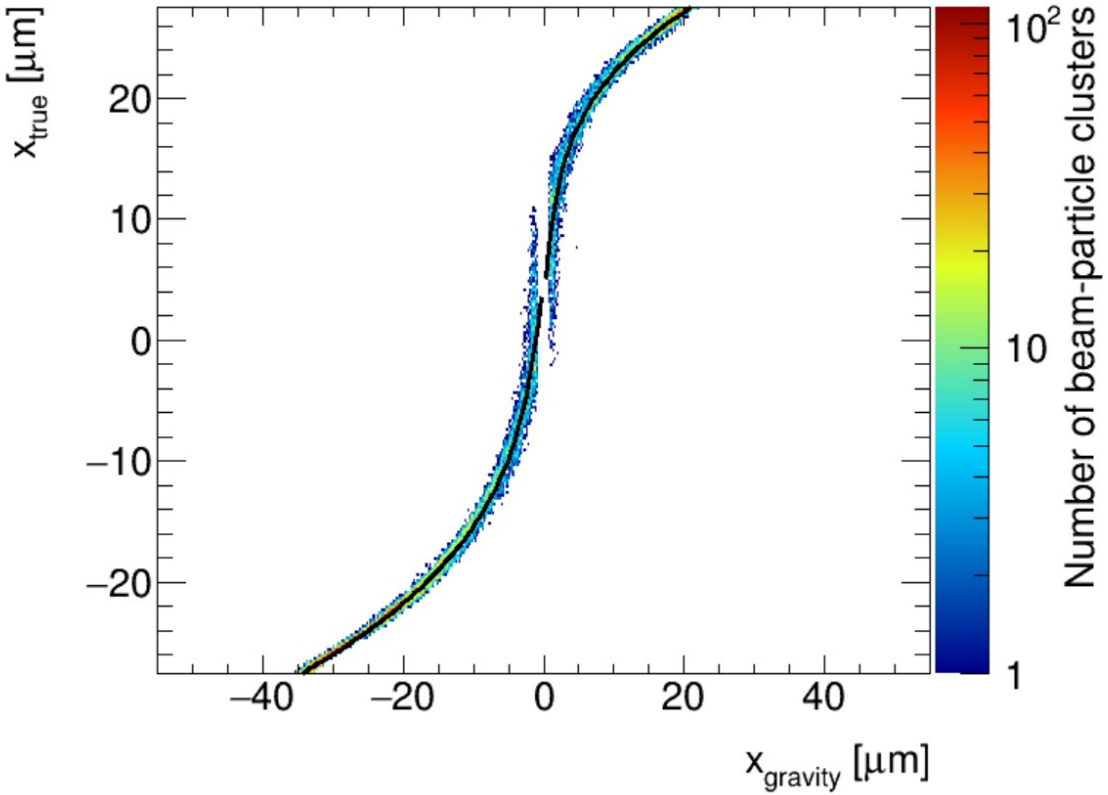
Clusters consisting of 3 or more pixels after X-axis projection



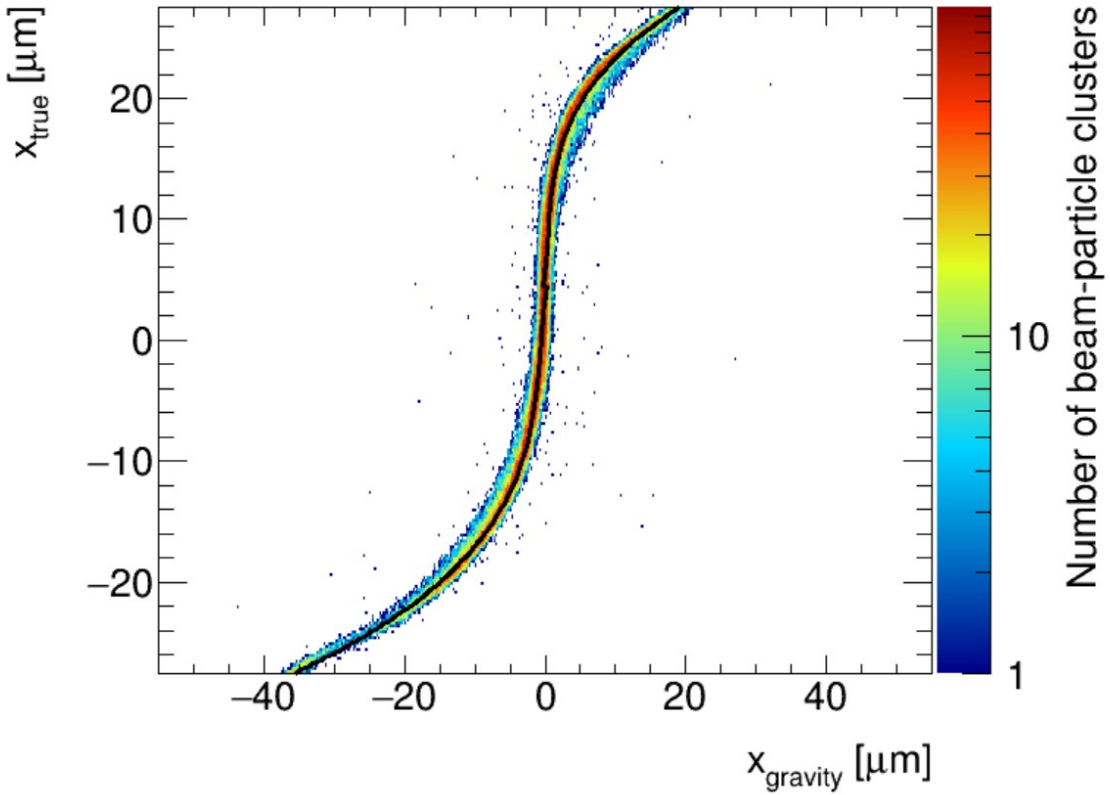
S-curves from ~sdoronin/public/Simulation_outfile_EnergyCalibration.root

Muons, dummy radiator, sides+corners

Clusters consisting of exactly 2 pixels after X-axis projection



Clusters consisting of 3 or more pixels after X-axis projection



S-curves from ~sdoronin/public/Simulation_outfile_EnergyCalibration.root

Muons, dummy radiator, sides+corners

Clusters consisting of exactly 2 pixels after X-axis projection

Formula for the left fit is

```
std::cbrt((2.*std::pow(-4.18364e-03/1.12142e-03,3)/27.- -4.18364e-03/1.12142e-03*2.45907e-01/1.12142e-03/3.+(-1.17690e+00-x)/1.12142e-03)/(-2.)+std::sqrt(TMath::Sq(2.*std::pow(-4.18364e-03/1.12142e-03,3)/27.- -4.18364e-03/1.12142e-03*2.45907e-01/1.12142e-03/3.+(-1.17690e+00-x)/1.12142e-03)/4.+std::pow(-TMath::Sq(-4.18364e-03/1.12142e-03)/3.+2.45907e-01/1.12142e-03,3)/27.))
+std::cbrt((2.*std::pow(-4.18364e-03/1.12142e-03,3)/27.- -4.18364e-03/1.12142e-03*2.45907e-01/1.12142e-03/3.+(-1.17690e+00-x)/1.12142e-03)/(-2.)-std::sqrt(TMath::Sq(2.*std::pow(-4.18364e-03/1.12142e-03,3)/27.- -4.18364e-03/1.12142e-03*2.45907e-01/1.12142e-03/3.+(-1.17690e+00-x)/1.12142e-03)/4.+std::pow(-TMath::Sq(-4.18364e-03/1.12142e-03)/3.+2.45907e-01/1.12142e-03,3)/27.))- -4.18364e-03/1.12142e-03/3.
```

Formula for the right fit is

```
std::cbrt((2.*std::pow(-4.64513e-02/2.05140e-03,3)/27.- -4.64513e-02/2.05140e-03*5.43923e-01/2.05140e-03/3.+(-1.55088e+00-x)/2.05140e-03)/(-2.)+std::sqrt(TMath::Sq(2.*std::pow(-4.64513e-02/2.05140e-03,3)/27.- -4.64513e-02/2.05140e-03*5.43923e-01/2.05140e-03/3.+(-1.55088e+00-x)/2.05140e-03)/4.+std::pow(-TMath::Sq(-4.64513e-02/2.05140e-03)/3.+5.43923e-01/2.05140e-03,3)/27.))
+std::cbrt((2.*std::pow(-4.64513e-02/2.05140e-03,3)/27.- -4.64513e-02/2.05140e-03*5.43923e-01/2.05140e-03/3.+(-1.55088e+00-x)/2.05140e-03)/(-2.)-std::sqrt(TMath::Sq(2.*std::pow(-4.64513e-02/2.05140e-03,3)/27.- -4.64513e-02/2.05140e-03*5.43923e-01/2.05140e-03/3.+(-1.55088e+00-x)/2.05140e-03)/4.+std::pow(-TMath::Sq(-4.64513e-02/2.05140e-03)/3.+5.43923e-01/2.05140e-03,3)/27.))- -4.64513e-02/2.05140e-03/3.
```

Clusters consisting of 3 or more pixels after X-axis projection

Formula for the left fit is

```
std::cbrt((2.*std::pow(3.60135e-03/1.63103e-03,3)/27.-3.60135e-03/1.63103e-03*1.42519e-01/1.63103e-03/3.+(-5.75728e-01-x)/1.63103e-03)/(-2.)+std::sqrt(TMath::Sq(2.*std::pow(3.60135e-03/1.63103e-03,3)/27.-3.60135e-03/1.63103e-03*1.42519e-01/1.63103e-03/3.+(-5.75728e-01-x)/1.63103e-03)/4.+std::pow(-TMath::Sq(3.60135e-03/1.63103e-03)/3.+1.42519e-01/1.63103e-03,3)/27.))
+std::cbrt((2.*std::pow(3.60135e-03/1.63103e-03,3)/27.-3.60135e-03/1.63103e-03*1.42519e-01/1.63103e-03/3.+(-5.75728e-01-x)/1.63103e-03)/(-2.)-std::sqrt(TMath::Sq(2.*std::pow(3.60135e-03/1.63103e-03,3)/27.-3.60135e-03/1.63103e-03*1.42519e-01/1.63103e-03/3.+(-5.75728e-01-x)/1.63103e-03)/4.+std::pow(-TMath::Sq(3.60135e-03/1.63103e-03)/3.+1.42519e-01/1.63103e-03,3)/27.))-3.60135e-03/1.63103e-03/3.
```

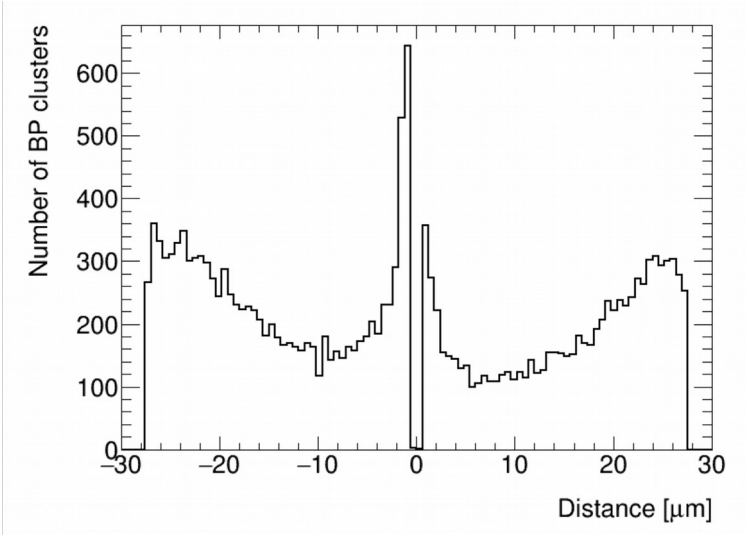
Formula for the right fit is

```
std::cbrt((2.*std::pow(-6.14151e-02/2.40612e-03,3)/27.- -6.14151e-02/2.40612e-03*6.41098e-01/2.40612e-03/3.+(-2.13525e+00-x)/2.40612e-03)/(-2.)+std::sqrt(TMath::Sq(2.*std::pow(-6.14151e-02/2.40612e-03,3)/27.- -6.14151e-02/2.40612e-03*6.41098e-01/2.40612e-03/3.+(-2.13525e+00-x)/2.40612e-03)/4.+std::pow(-TMath::Sq(-6.14151e-02/2.40612e-03)/3.+6.41098e-01/2.40612e-03,3)/27.))
+std::cbrt((2.*std::pow(-6.14151e-02/2.40612e-03,3)/27.- -6.14151e-02/2.40612e-03*6.41098e-01/2.40612e-03/3.+(-2.13525e+00-x)/2.40612e-03)/(-2.)-std::sqrt(TMath::Sq(2.*std::pow(-6.14151e-02/2.40612e-03,3)/27.- -6.14151e-02/2.40612e-03*6.41098e-01/2.40612e-03/3.+(-2.13525e+00-x)/2.40612e-03)/4.+std::pow(-TMath::Sq(-6.14151e-02/2.40612e-03)/3.+6.41098e-01/2.40612e-03,3)/27.))- -6.14151e-02/2.40612e-03/3.
```

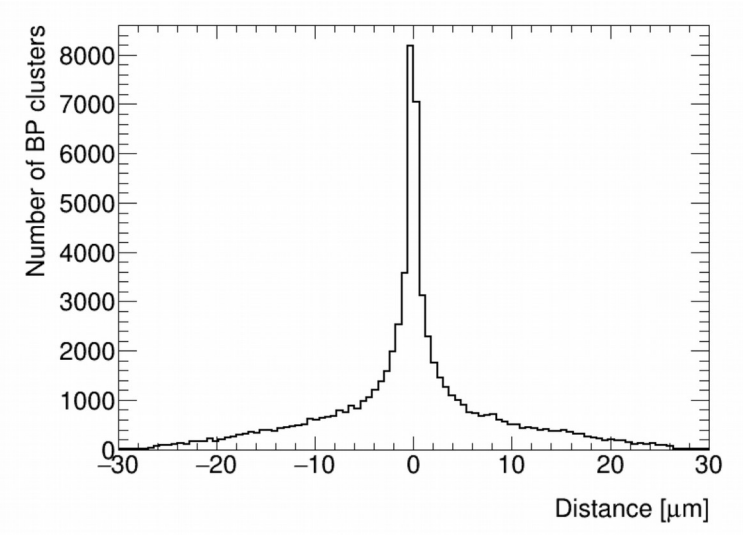
X-axis distances between the center of the most energetic pixel in a BP cluster and the center of this BP cluster: **before** S-curve-based correction

data, run 7, sides+corners

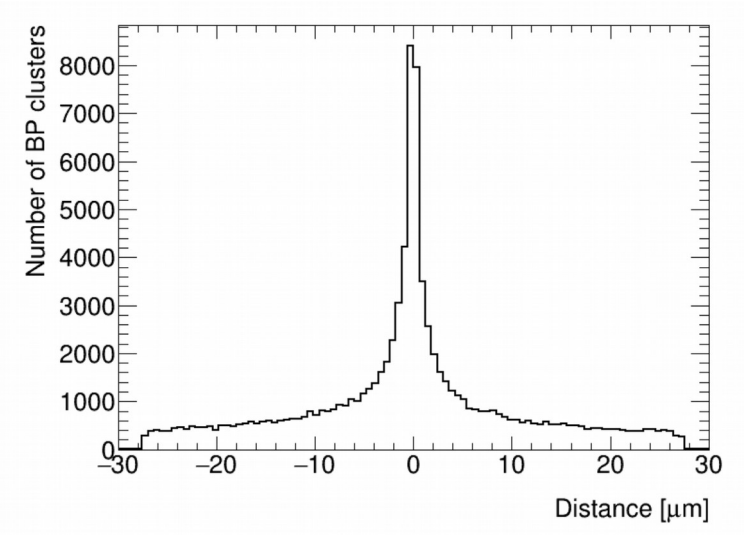
Clusters consisting of 2 pixels after X-axis projection



Clusters consisting of 3 or more pixels after X-axis projection



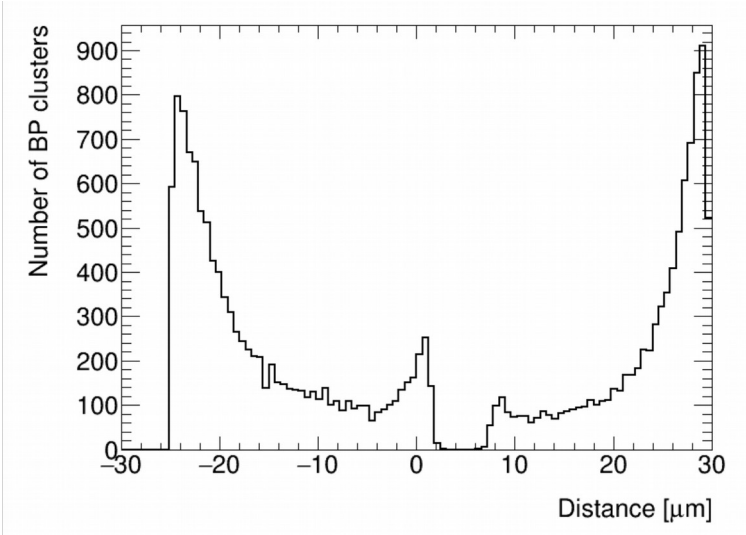
Clusters consisting of any number of pixels (even including 1 after X-axis projection)



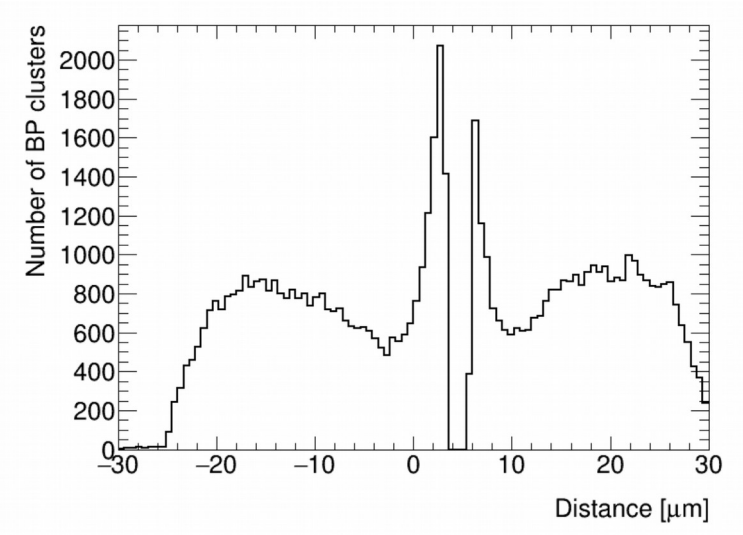
X-axis distances between the center of the most energetic pixel in a BP cluster and the center of this BP cluster: **after** S-curve-based correction

data, run 7, sides+corners

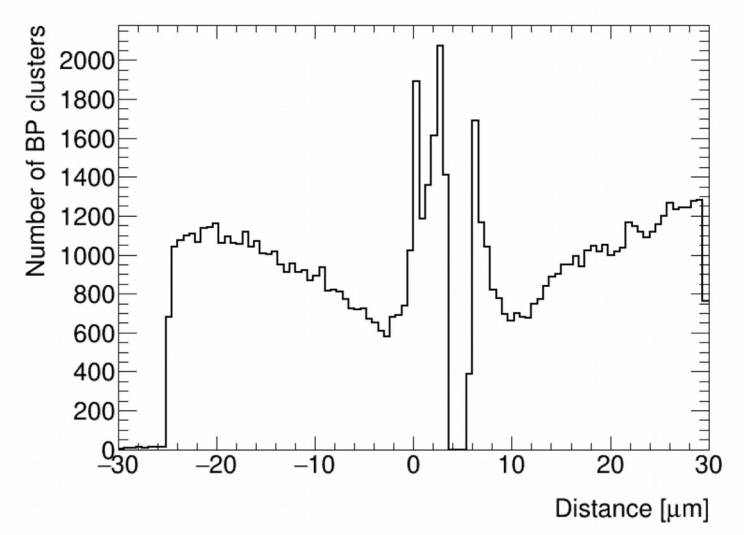
Clusters consisting of 2 pixels after X-axis projection



Clusters consisting of 3 or more pixels after X-axis projection



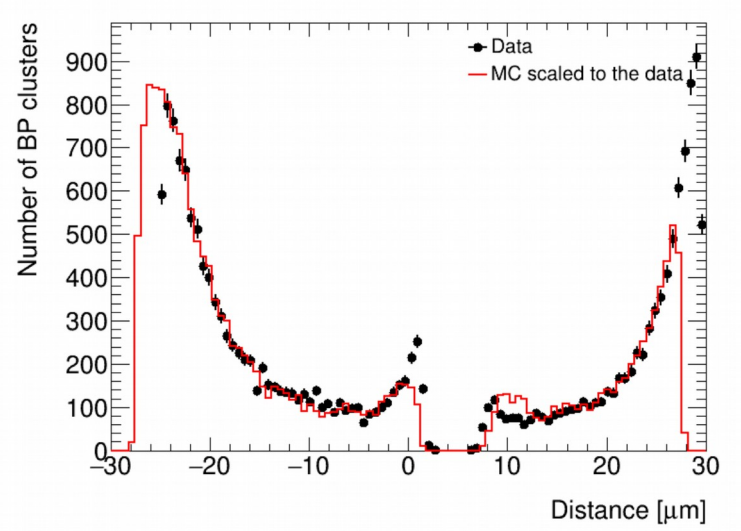
Clusters consisting of any number of pixels (even including 1 after X-axis projection)



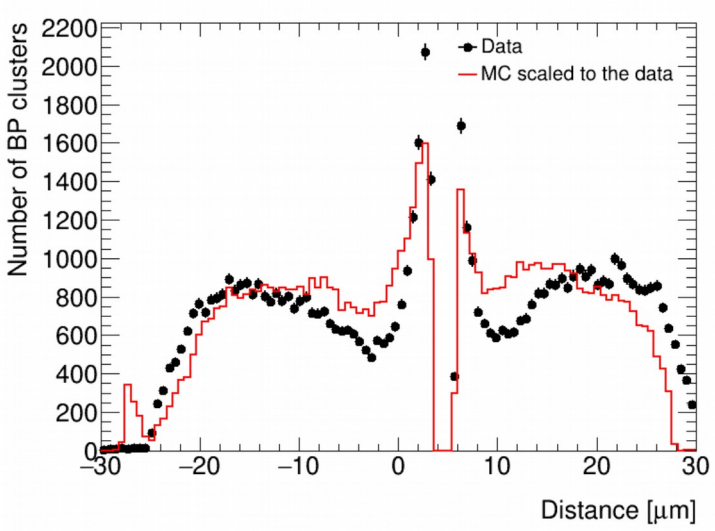
X-axis distances between the center of the most energetic pixel in a BP cluster and the center of this BP cluster: **after** S-curve-based correction

data vs MC, sides+corners

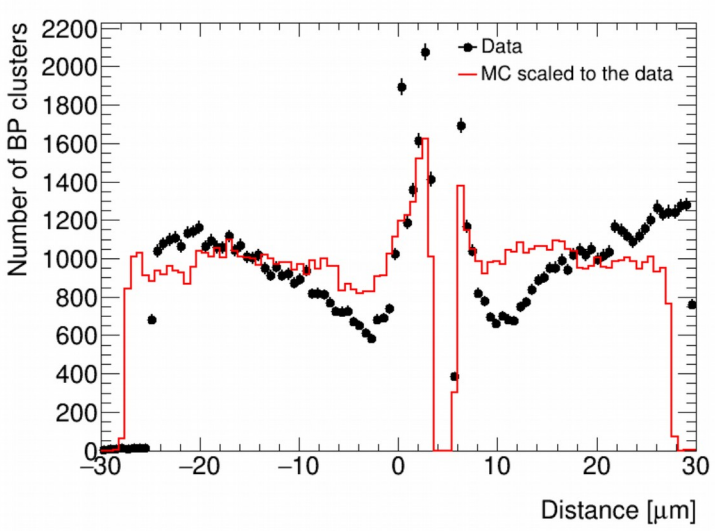
Clusters consisting of 2 pixels after X-axis projection



Clusters consisting of 3 or more pixels after X-axis projection



Clusters consisting of any number of pixels (even including 1 after X-axis projection)



Thank you for your attention!