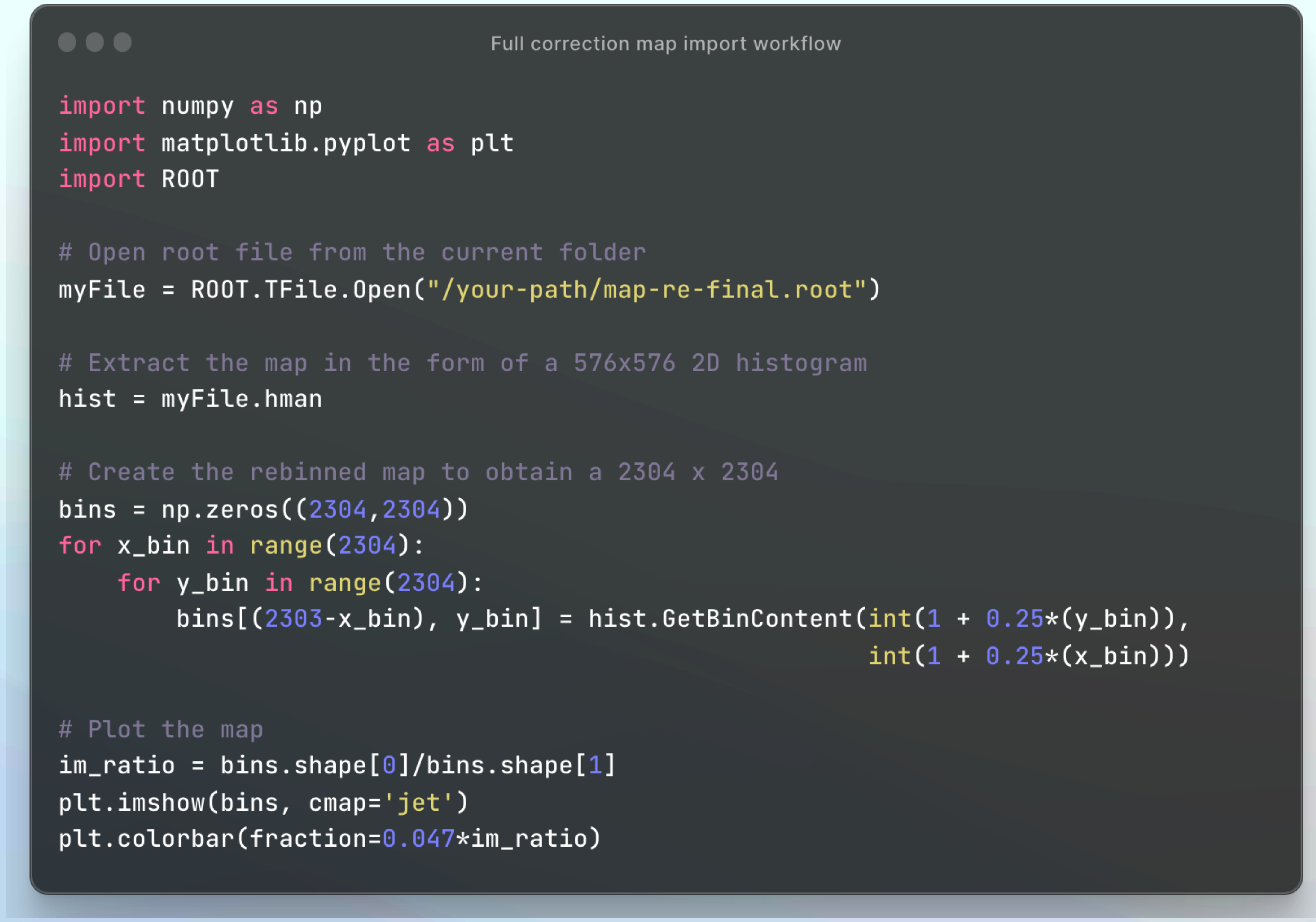
# Testing the disuniformity map goodness

Luca Zappaterra, Davide Pinci, 6th March 2024

# Importing the map

- From the Github page you can retrieve the instructions to apply Davide’s disuniformity map in ROOT.
- To use it in Python, one should deal with the **different indexing conventions** implemented between Python and ROOT, together with the **inversion of the “x” axis** and the **rebinning**.



Full correction map import workflow

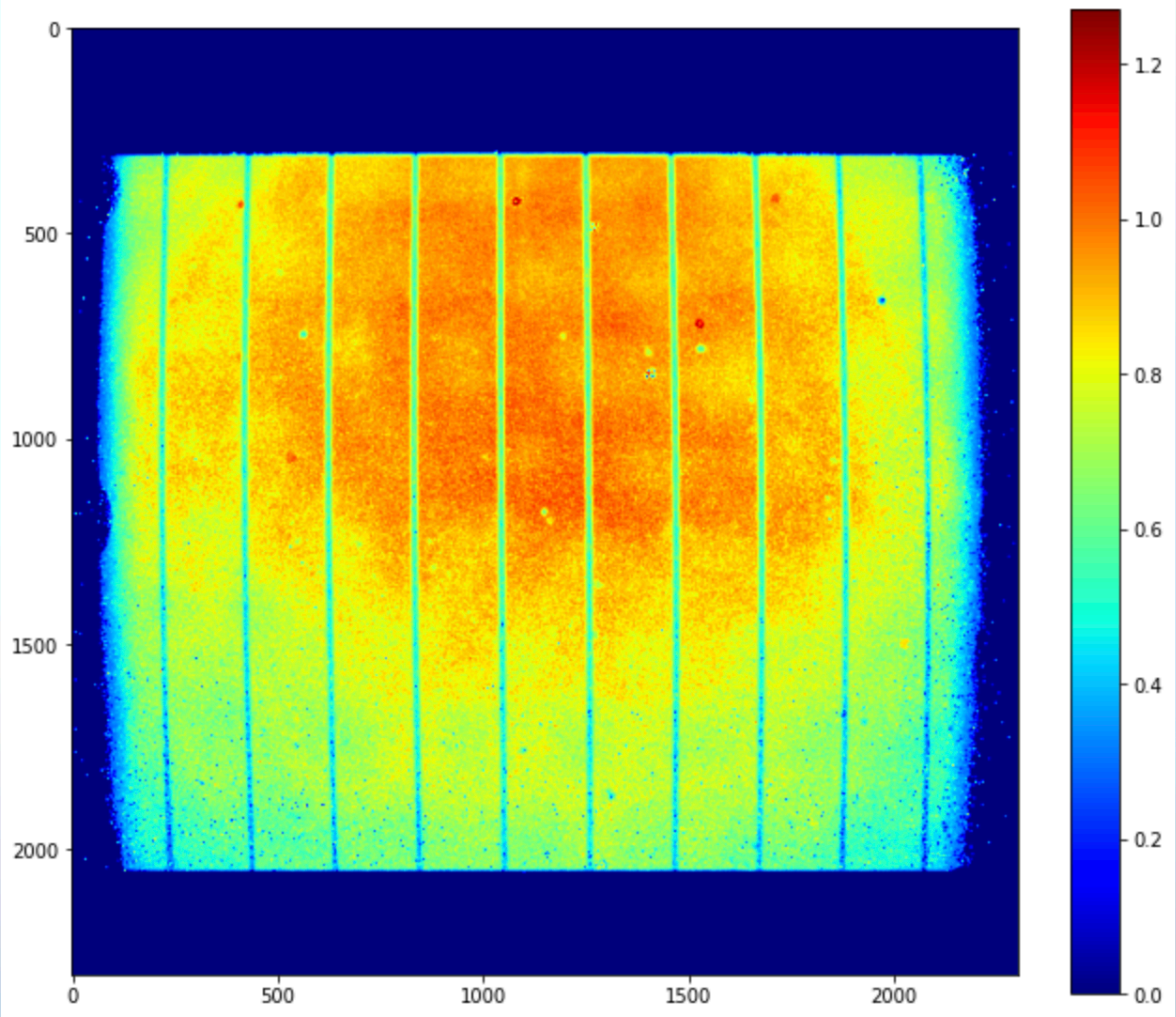
```python
import numpy as np
import matplotlib.pyplot as plt
import ROOT

# Open root file from the current folder
myFile = ROOT.TFile.Open("/your-path/map-re-final.root")

# Extract the map in the form of a 576x576 2D histogram
hist = myFile.hman

# Create the rebinned map to obtain a 2304 x 2304
bins = np.zeros((2304,2304))
for x_bin in range(2304):
    for y_bin in range(2304):
        bins[(2303-x_bin), y_bin] = hist.GetBinContent(int(1 + 0.25*(y_bin)),
                                                       int(1 + 0.25*(x_bin)))

# Plot the map
im_ratio = bins.shape[0]/bins.shape[1]
plt.imshow(bins, cmap='jet')
plt.colorbar(fraction=0.047*im_ratio)
```



# Importing the map: an important remark

- **The 2D histogram** inside the root file **is a 578×578 matrix**, but in reality **we are not interested in the bins at the start and at the end**, filled with the overflow and underflow values.

- To remove them **we add the "1 +" inside the *GetBinContent()* function**, such that:

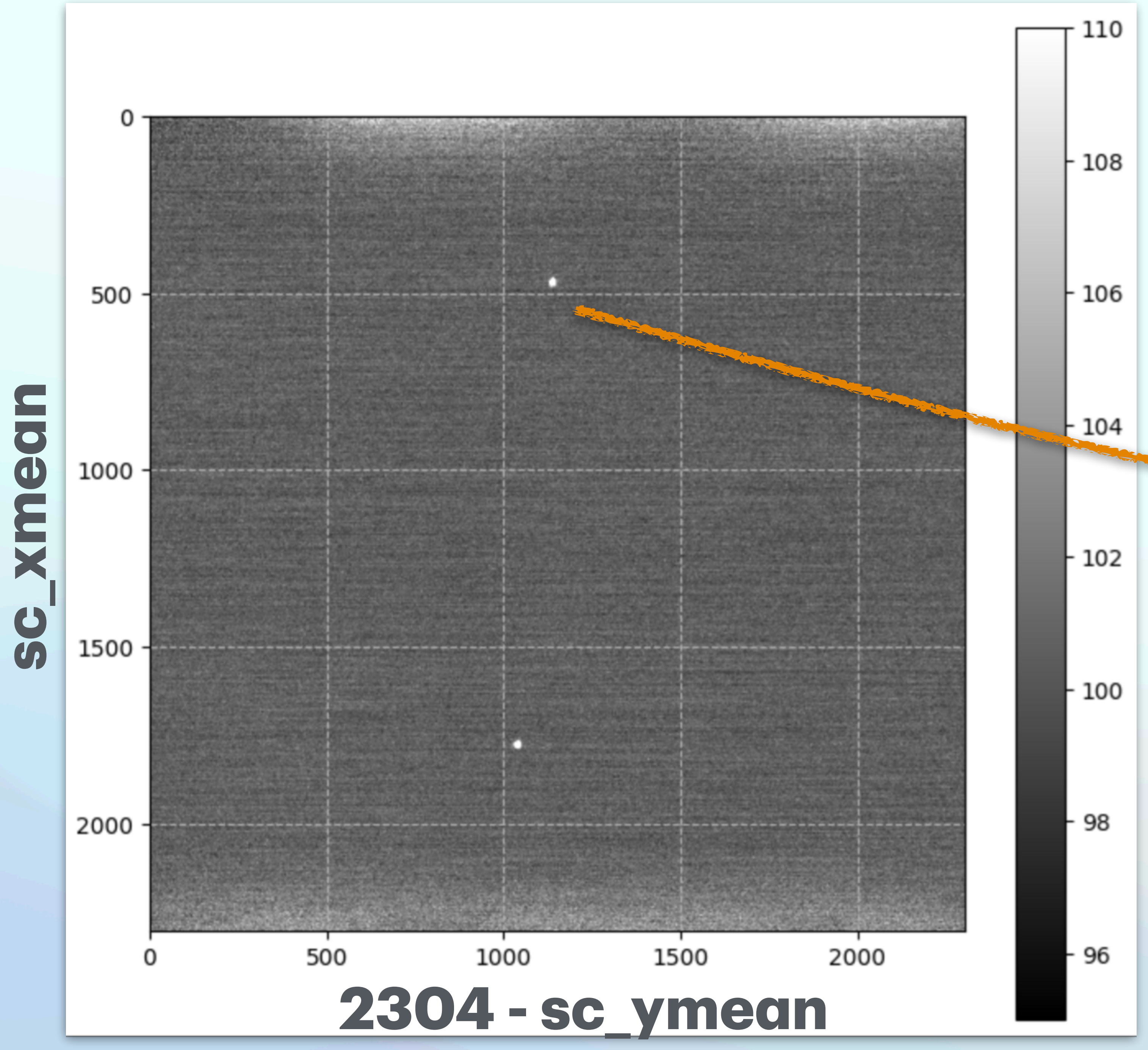
    - In 0 we get int(1 + 0.25*0) = 1, avoiding the underflow values in the bin (0,0).

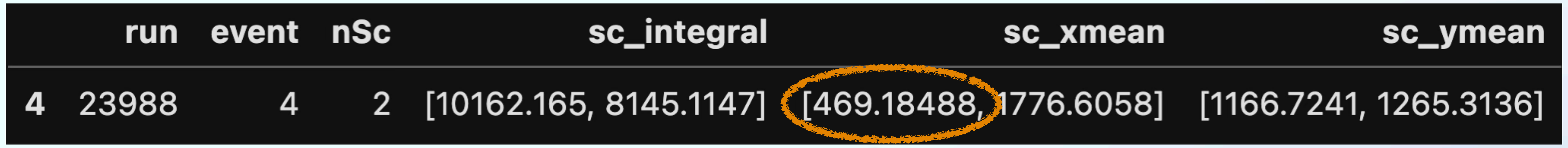
    - In 2303 we get int(1 + 0.25*2303) = 576, avoiding the overflow values in the bin (577, 577).

# Everything is a convention, let's set ours.

- The map is computed wrt the cluster barycentre (denoted by sc_xmean and sc_ymean).

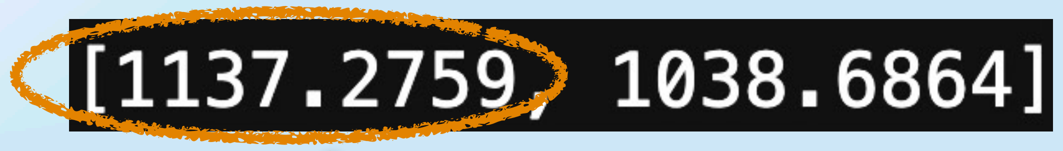
- Unfortunately, for Python what **we get as x_mean from the root file is what we usually call "y" and viceversa**, with the complication that **the x axis** (sc_ymean) **needs to be flipped** to match the image spatial coordinates.





| | run | event | nSc | sc_integral | sc_xmean | sc_ymean |
|---|---|---|---|---|---|---|
| **4** | 23988 | 4 | 2 | [10162.165, 8145.1147] | [469.18488, 1776.6058] | [1166.7241, 1265.3136] |

**2304 -**



```
[1137.2759, 1038.6864]
```

# Using the map

- Given what we saw in the previous slide, *the correct way to access the map after rebinning is to flip the sc_ymean coordinate from the root file, treating it as the x axis.*

- At this point the idea is that each cluster has a mid point and **one can loop over the clusters to correct its sc_integral by dividing it for the map value at the cluster's mid point**.

- You are free to implement it in the way you prefer, **the important thing is to flip sc_ymean**.

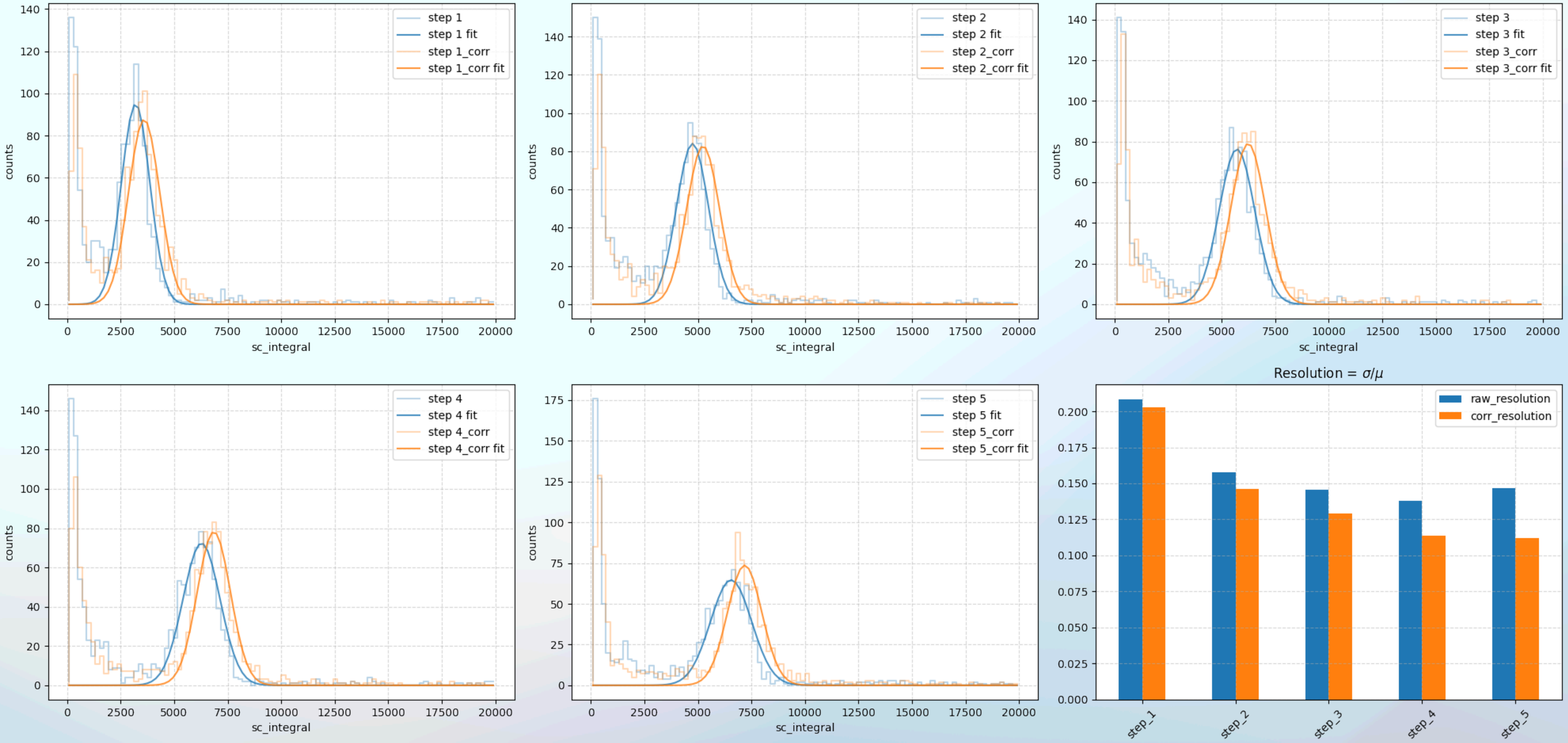
# Testing map on daily calibrations (May)

- To test the retroactive power of the disuniformity map we can test it on random daily scans from different months.

- We can fit the iron peak at each step in both **raw** and **map-corrected** case.

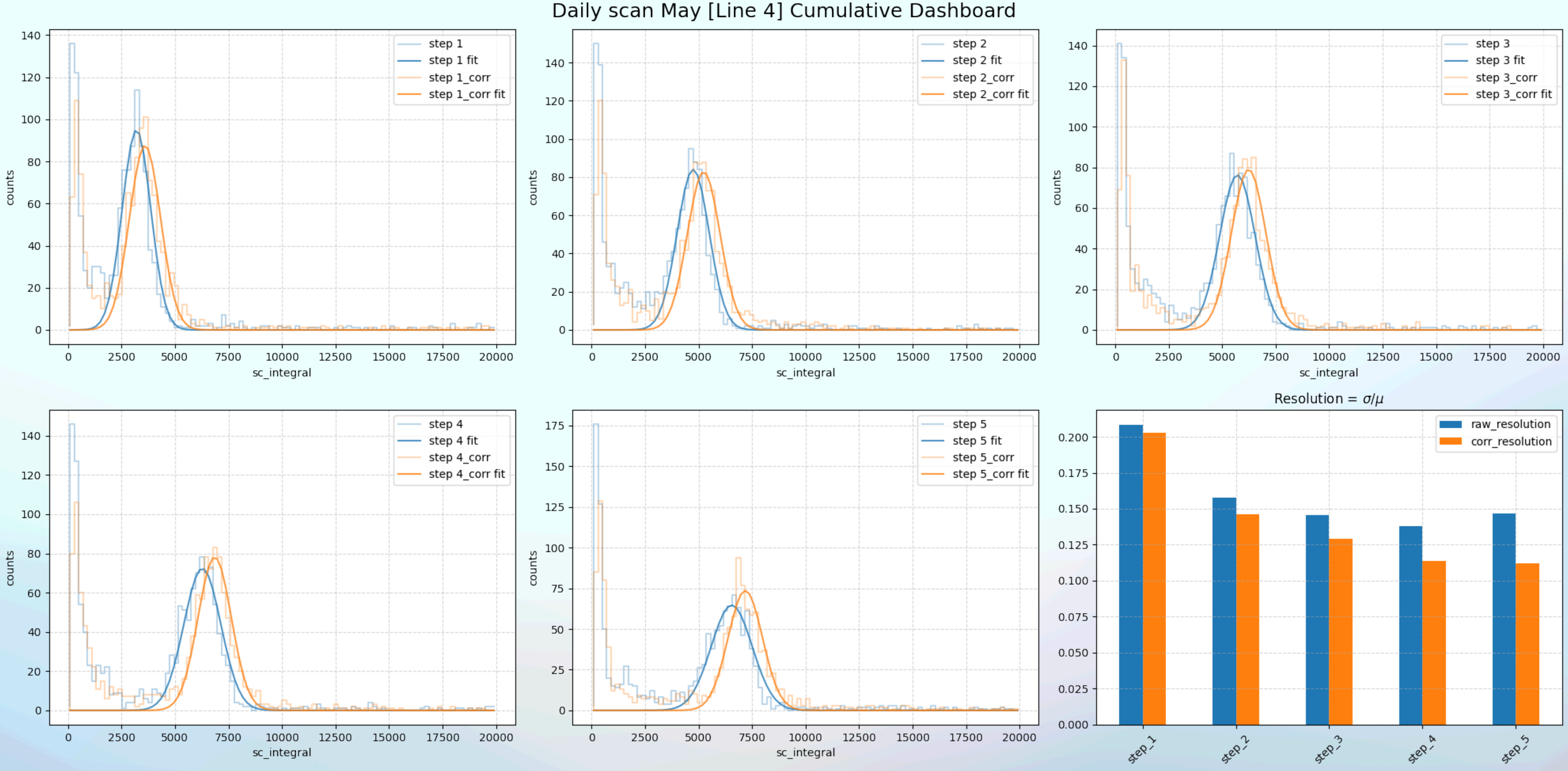
- From the deduced fit parameters **we can calculate the resolutions** ($\sigma/\mu$), which are always lower in the corrected case **(no cherry picking!)**

Daily scan May [Line 4] Cumulative Dashboard

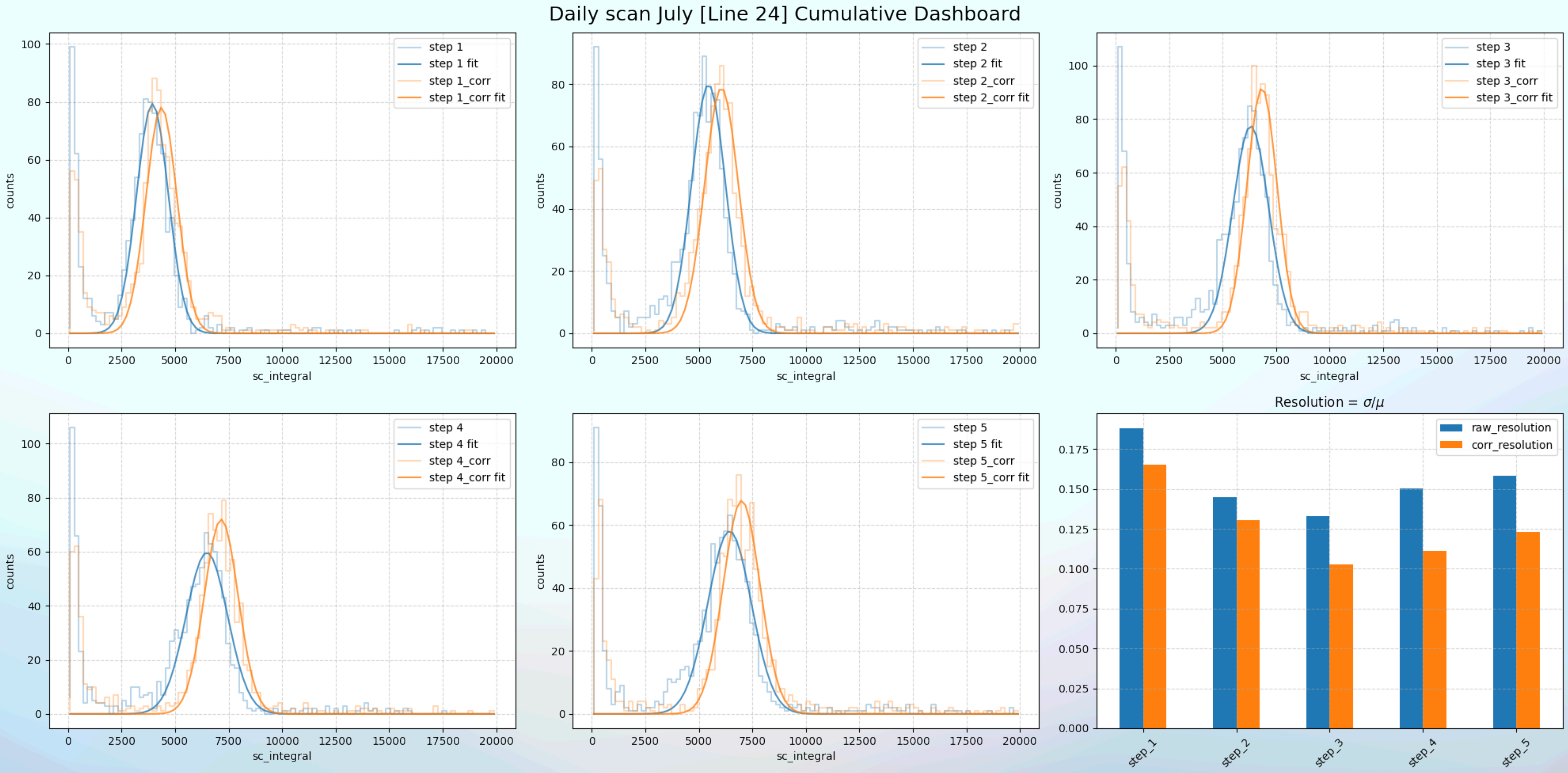


# Testing map on daily calibrations (May)

**SAME AS PREVIOUS, JUST TO COMPARE WITH THE FOLLOWINGS**



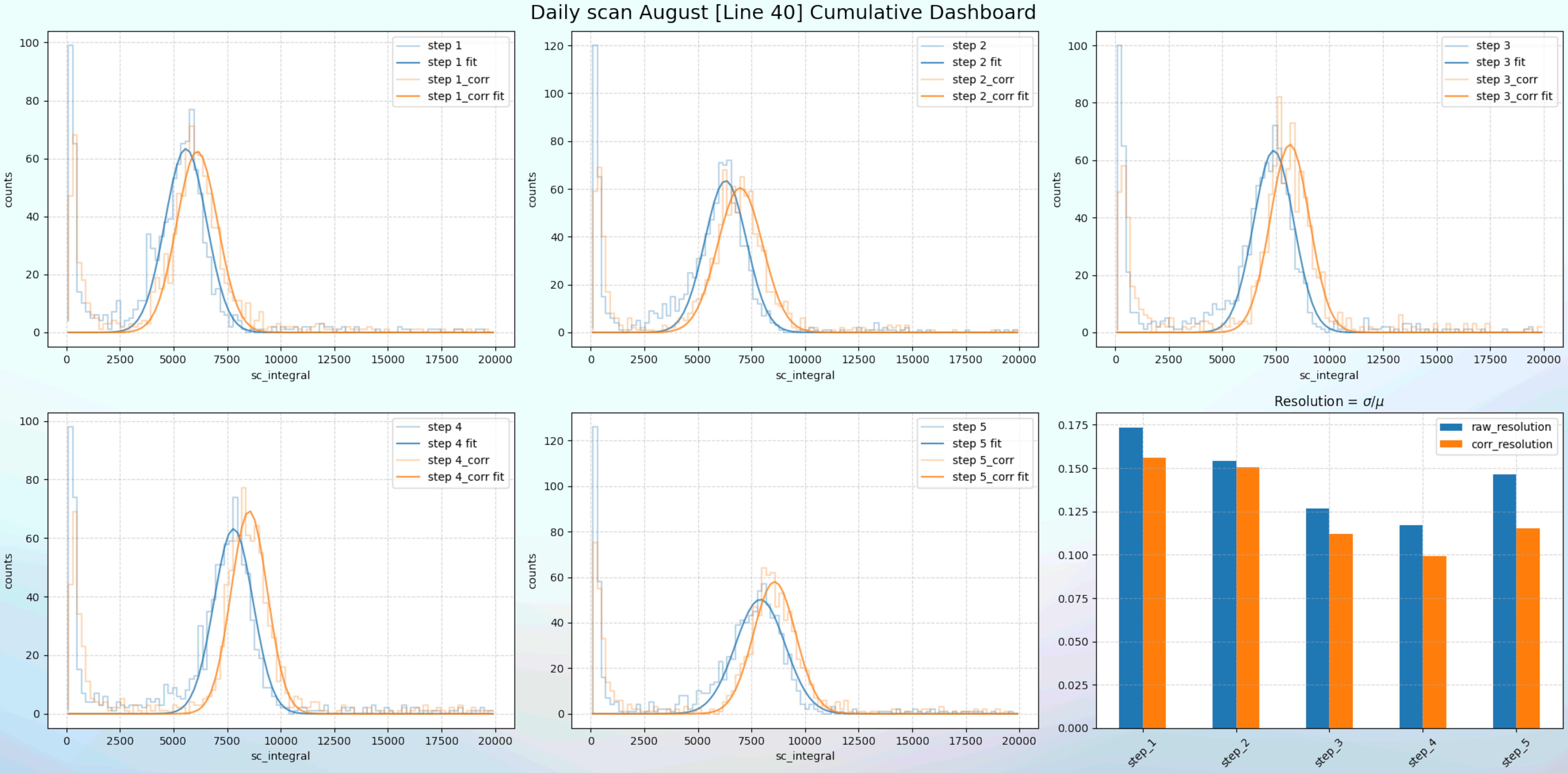
# Testing map on daily calibrations (July)



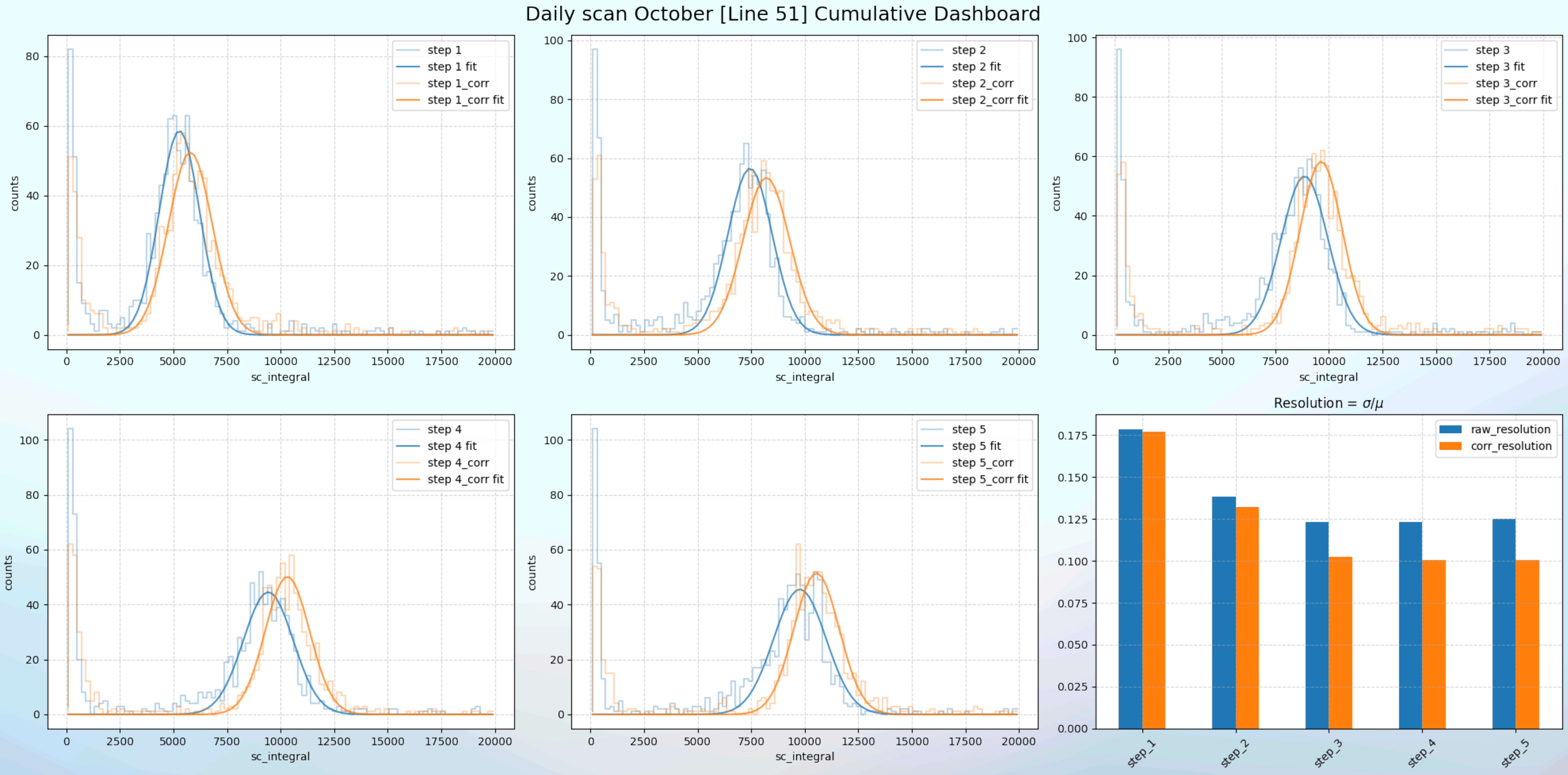
Daily scan July [Line 24] Cumulative Dashboard

Resolution = $\sigma/\mu$

# Testing map on daily calibrations (August)



# Testing map on daily calibrations (October)

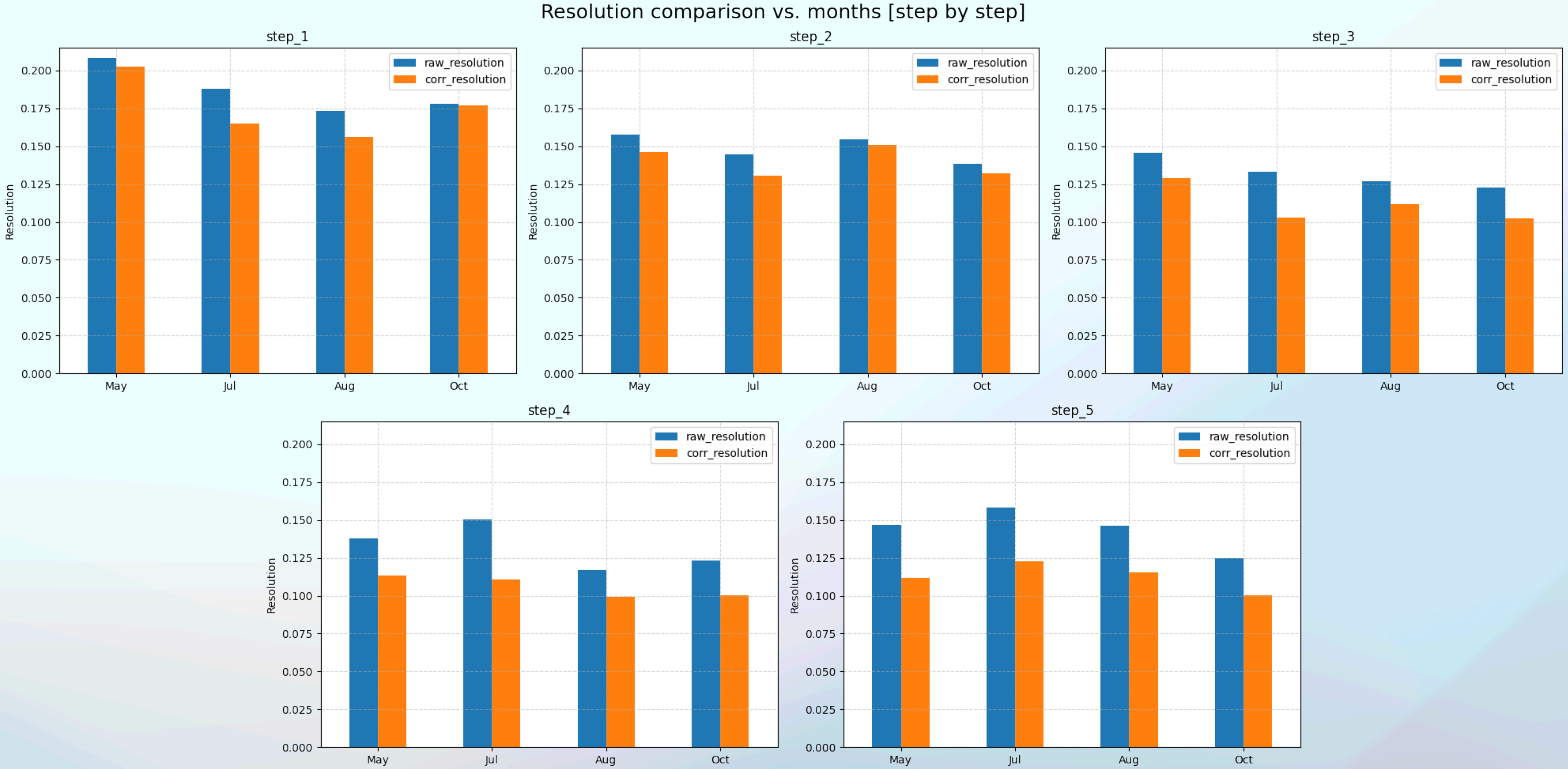


# Resolution comparison between months

- Here we can see the resolutions comparison between raw and map-corrected resolutions at every steps and at each month.

- *y-axis lim is fixed* to assure better comparison on the absolute value between steps.

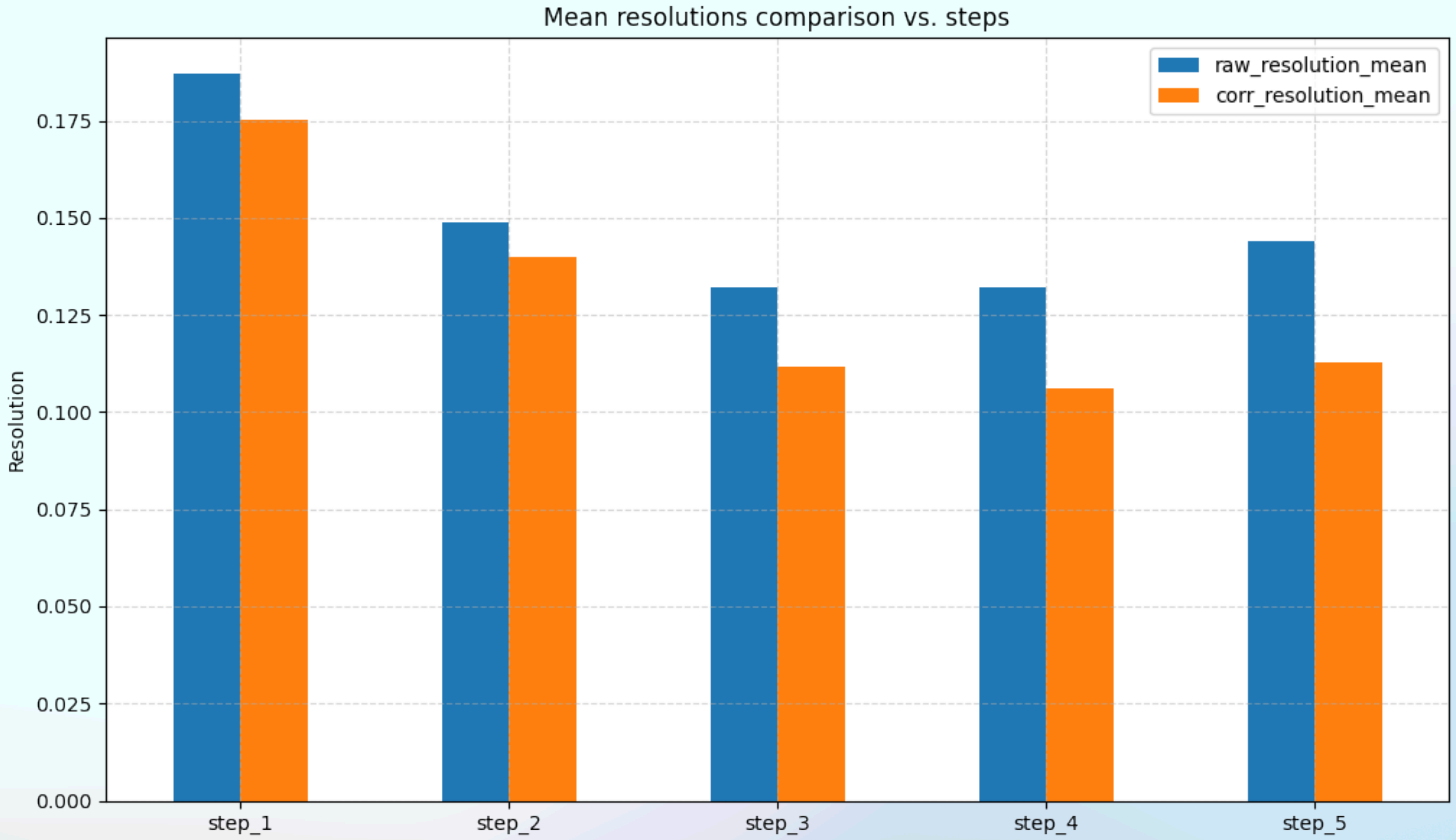
- This is **just a re-shuffle of the previous plots**, the results are the same.



# Mean resolution comparison between steps

- Here we can see the resolutions comparison between mean values (along months) of raw and map-corrected resolutions at every step.

- Of course the corrected resolution is still better than the raw one.



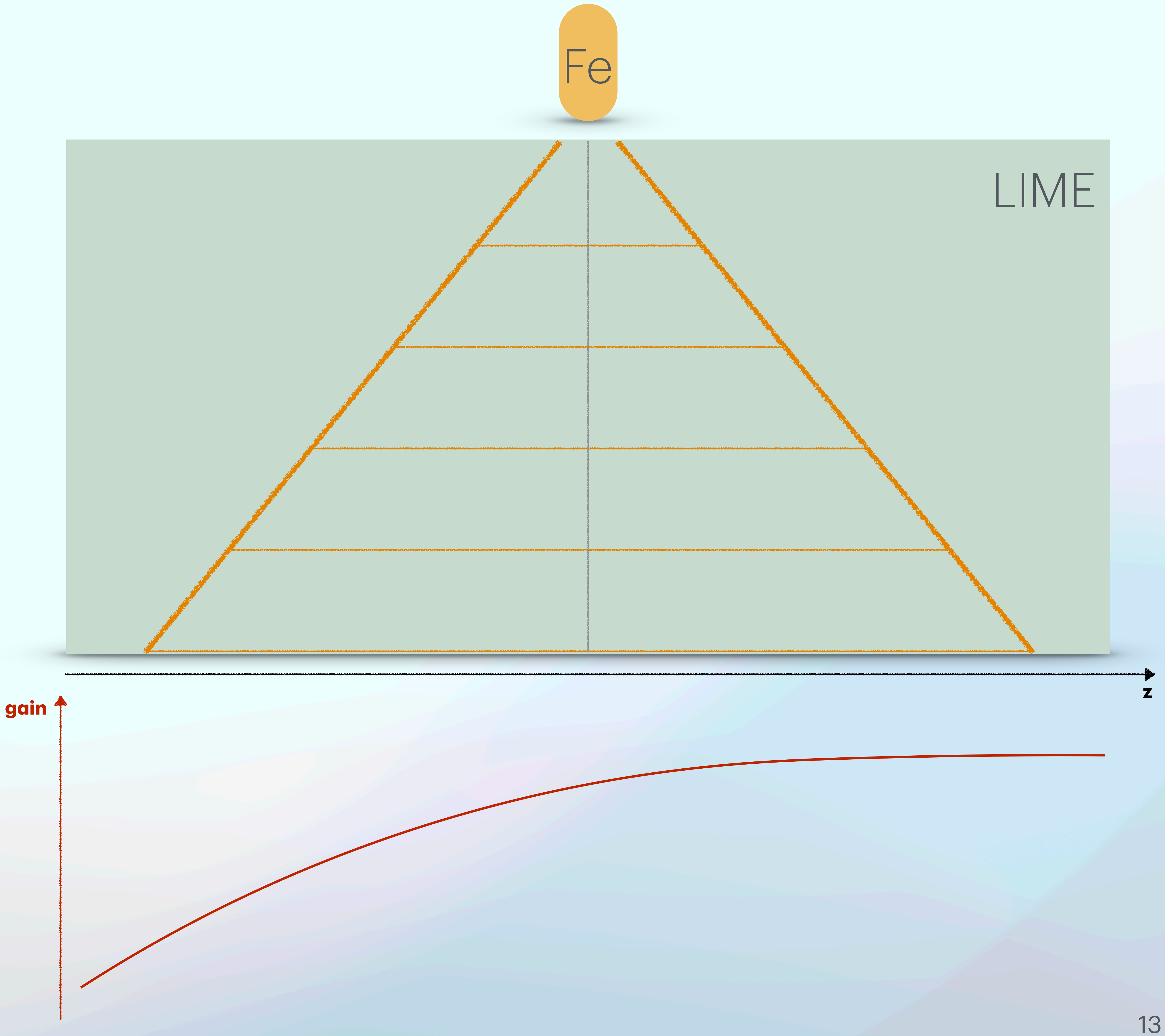
# Height scan

- Assume that we have an Iron source at step 3 and model the emission as a **rough cone**.

- Let’s slice the GEM plane into 5 different horizontal areas which extends in the z direction.

- We know that the gain scales with the z direction in a non-linear way.

- From the map we see a **disuniformity in the gain between the upper and lower part**. Can we explain it someway?



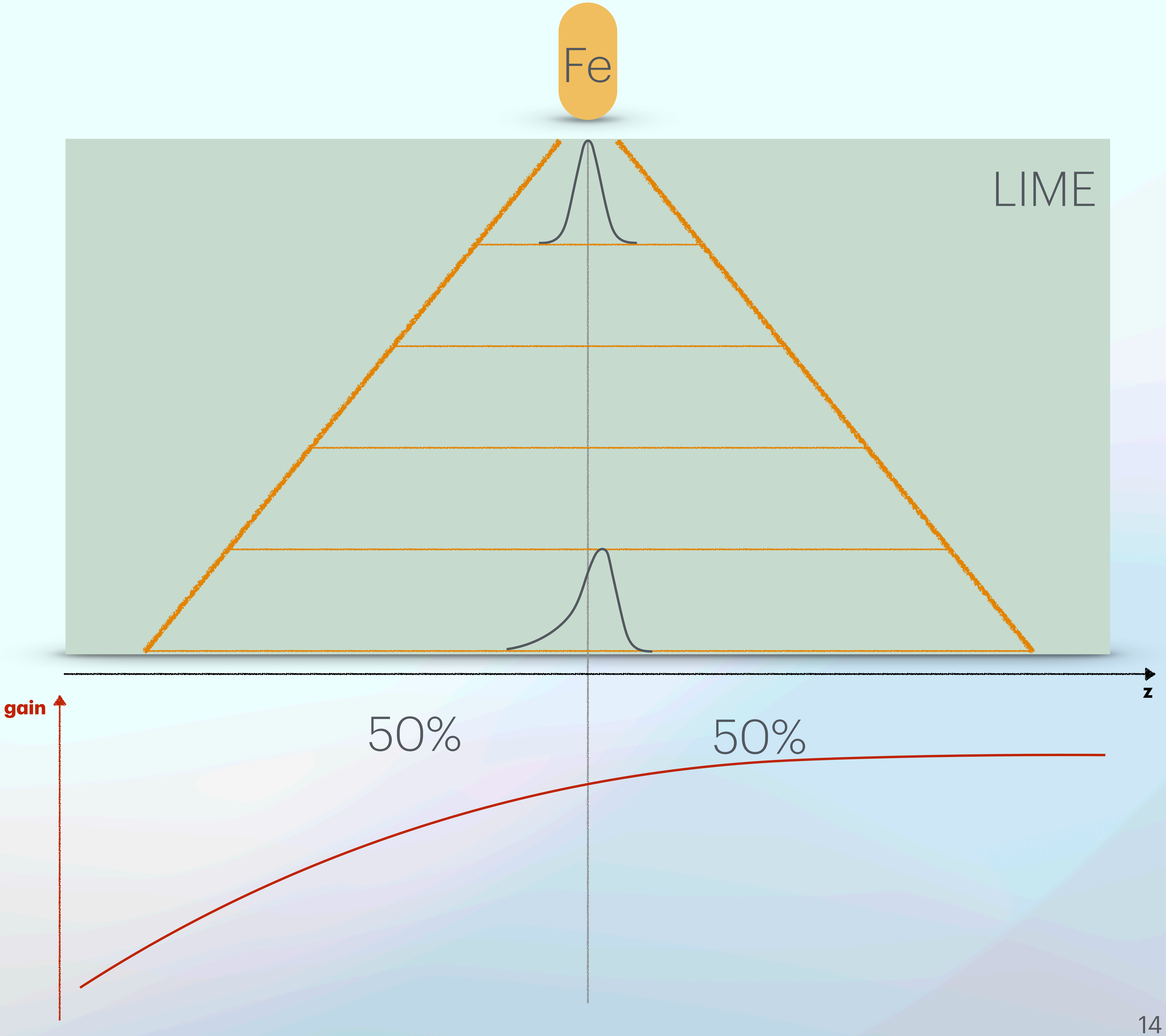
# Height scan

- Assume a 50/50 distribution for the events w.r.t the vertical line.

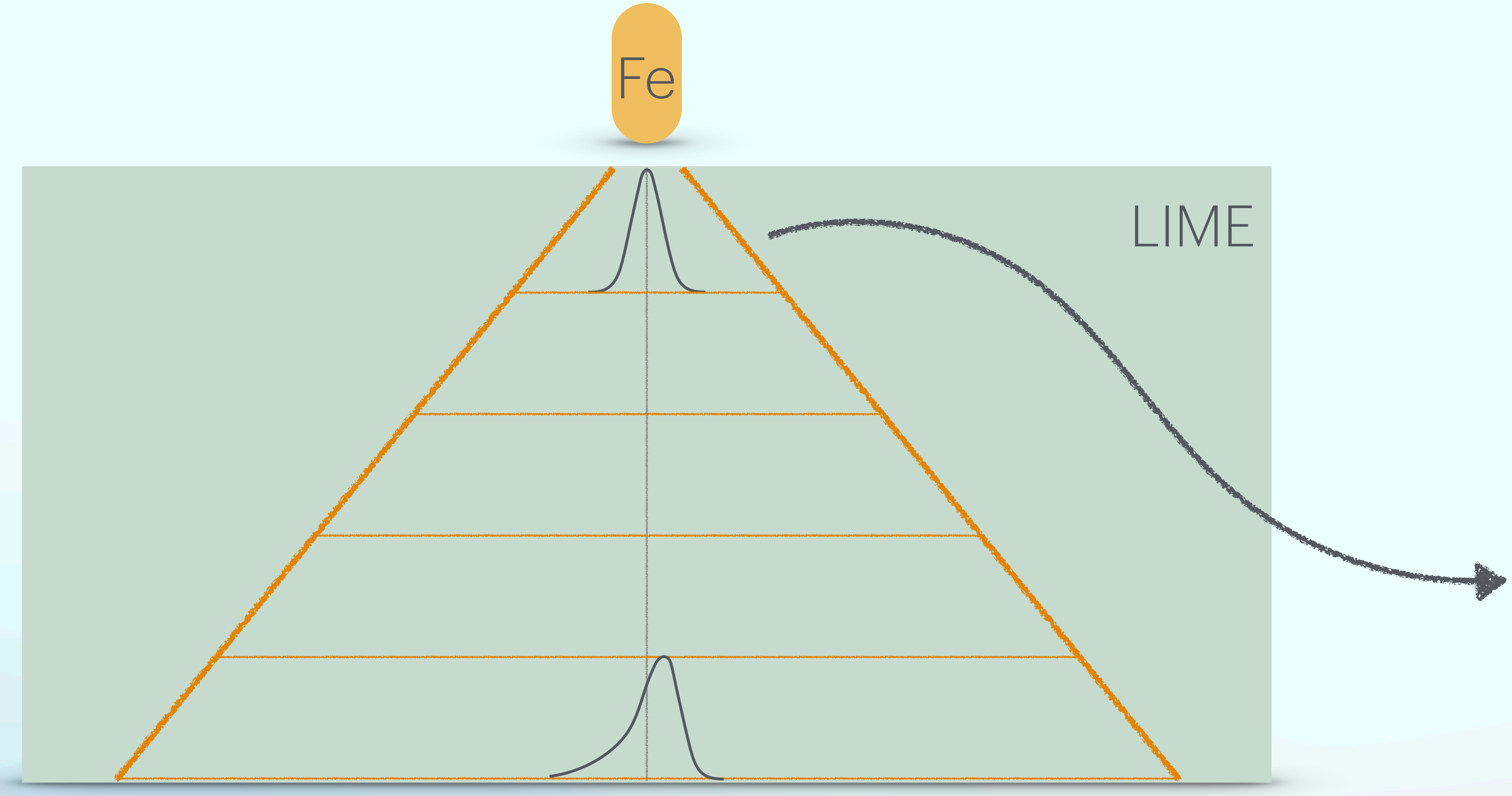
- This disuniformity could be explained from the fact that **upper regions are populated from events which are less spread in z**, hence the distribution is more gaussian.

- On the other hand, **lower regions are populated by events which are more spread in z** and are in turn affected by a large gain fluctuation.

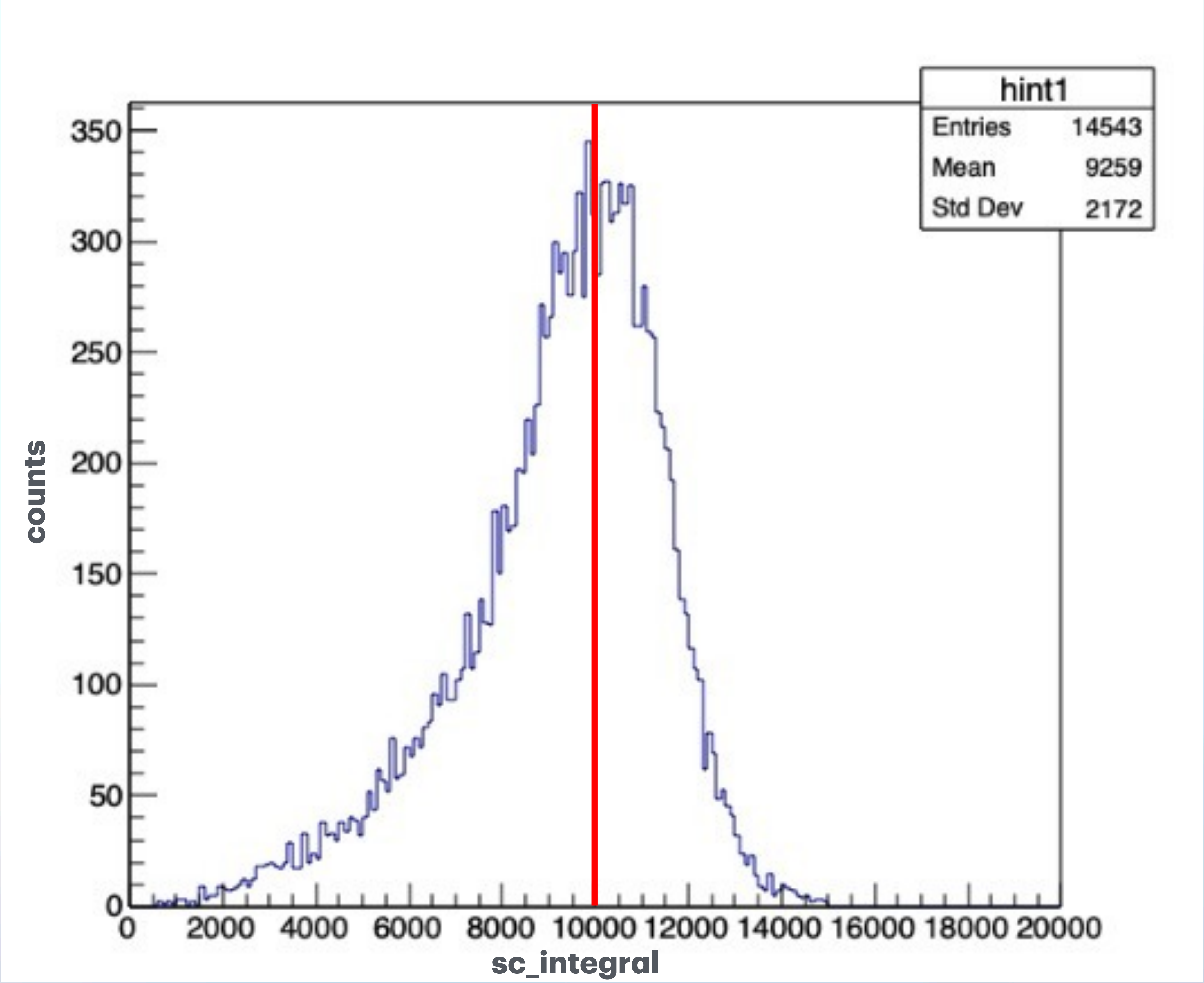
- Hence, **we expect** the *lower regions peak* to be:
  - *Shifted forward* from the high-z events
  - *With a left tail* produced from the low-z events



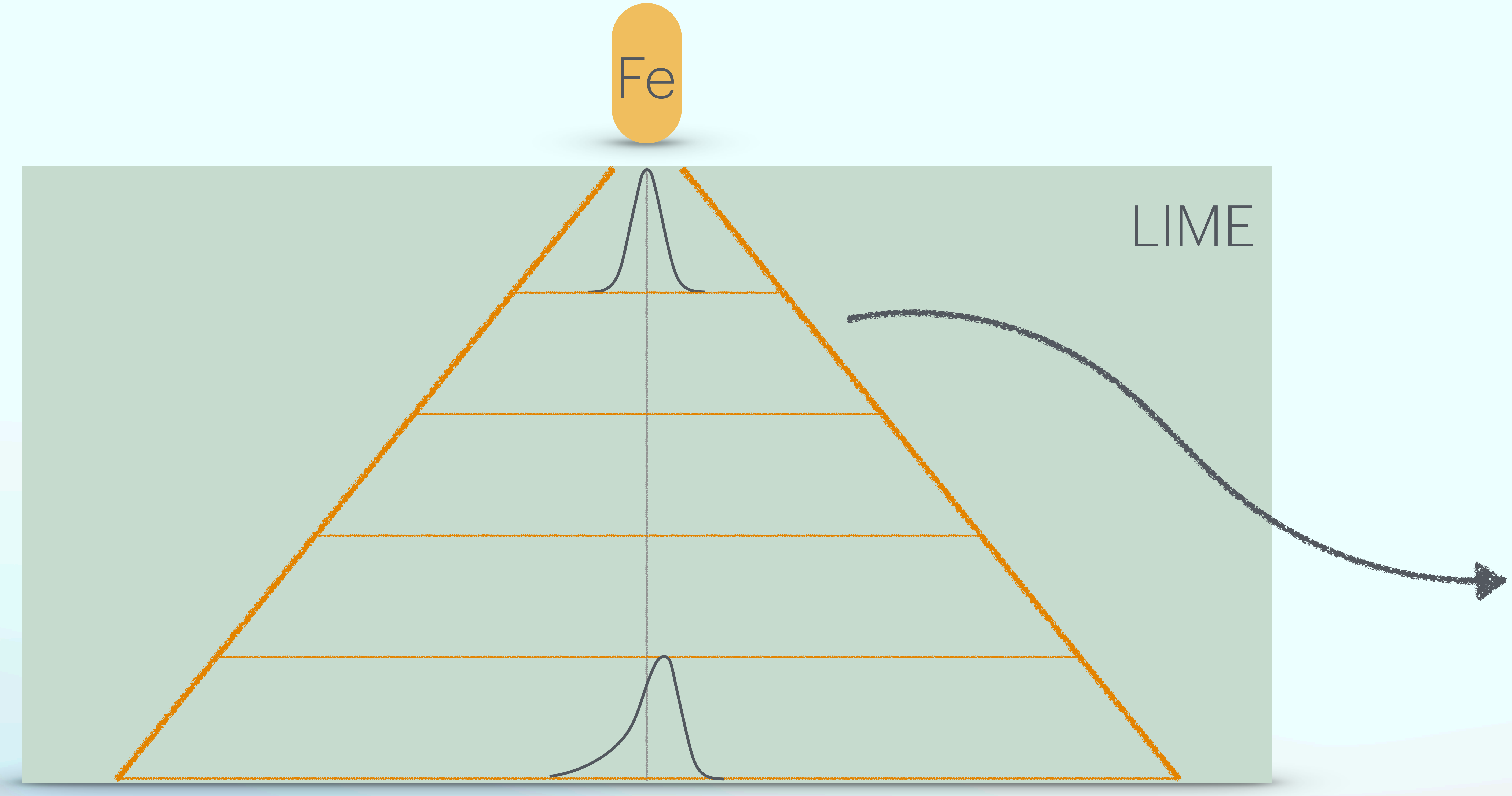
# Height scan - Observed data



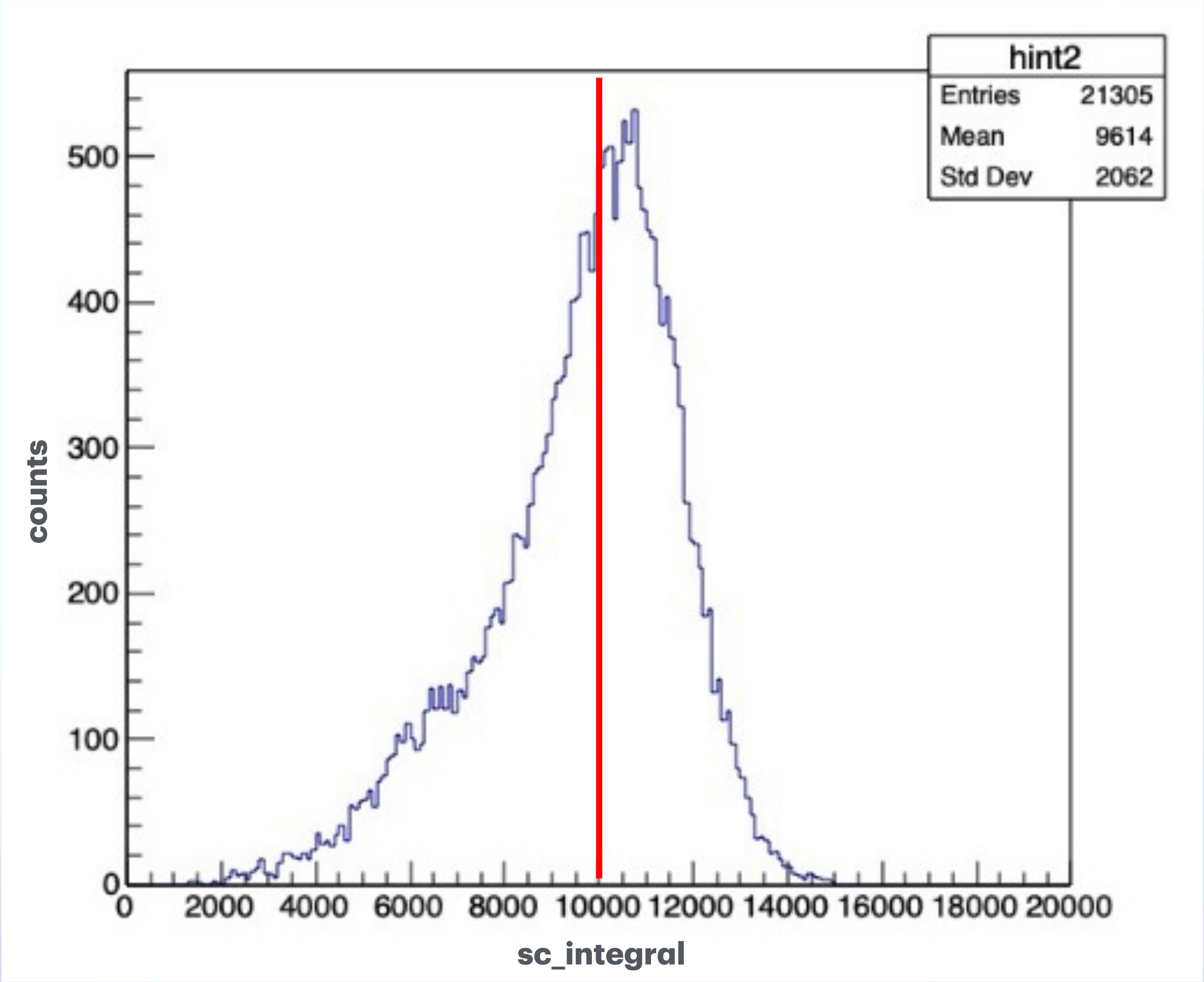
- First slice



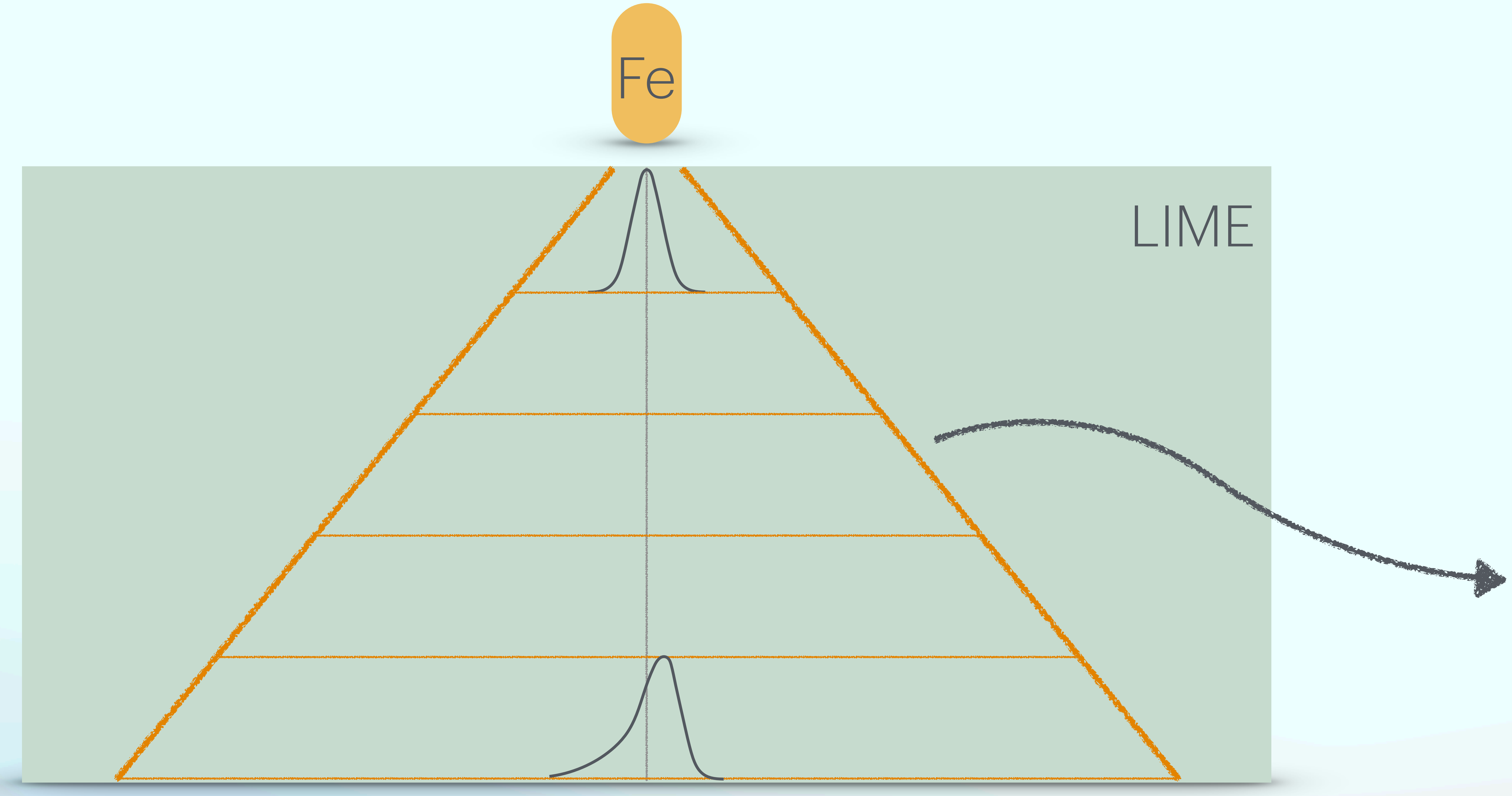
# Height scan - Observed data



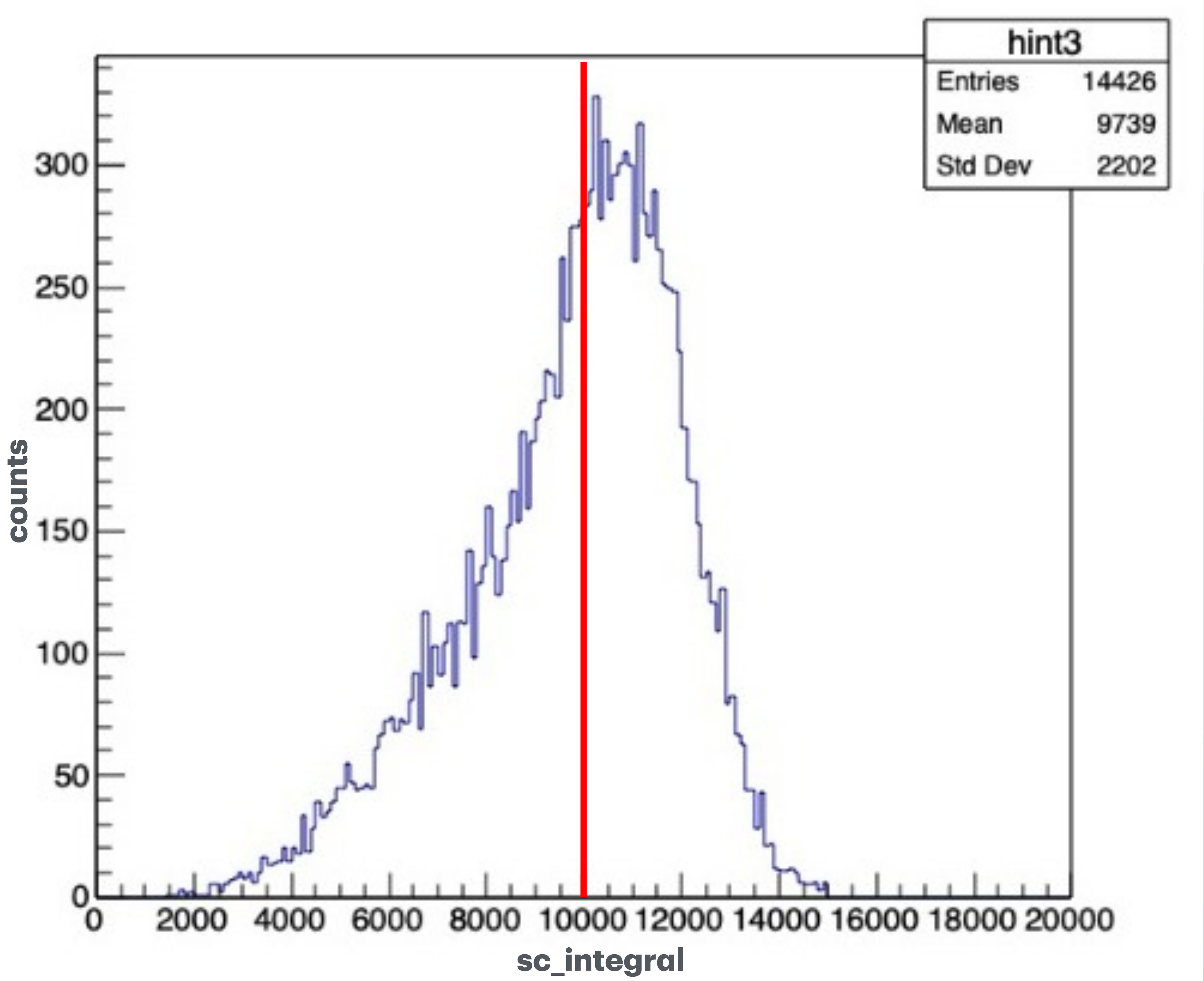
- Second slice



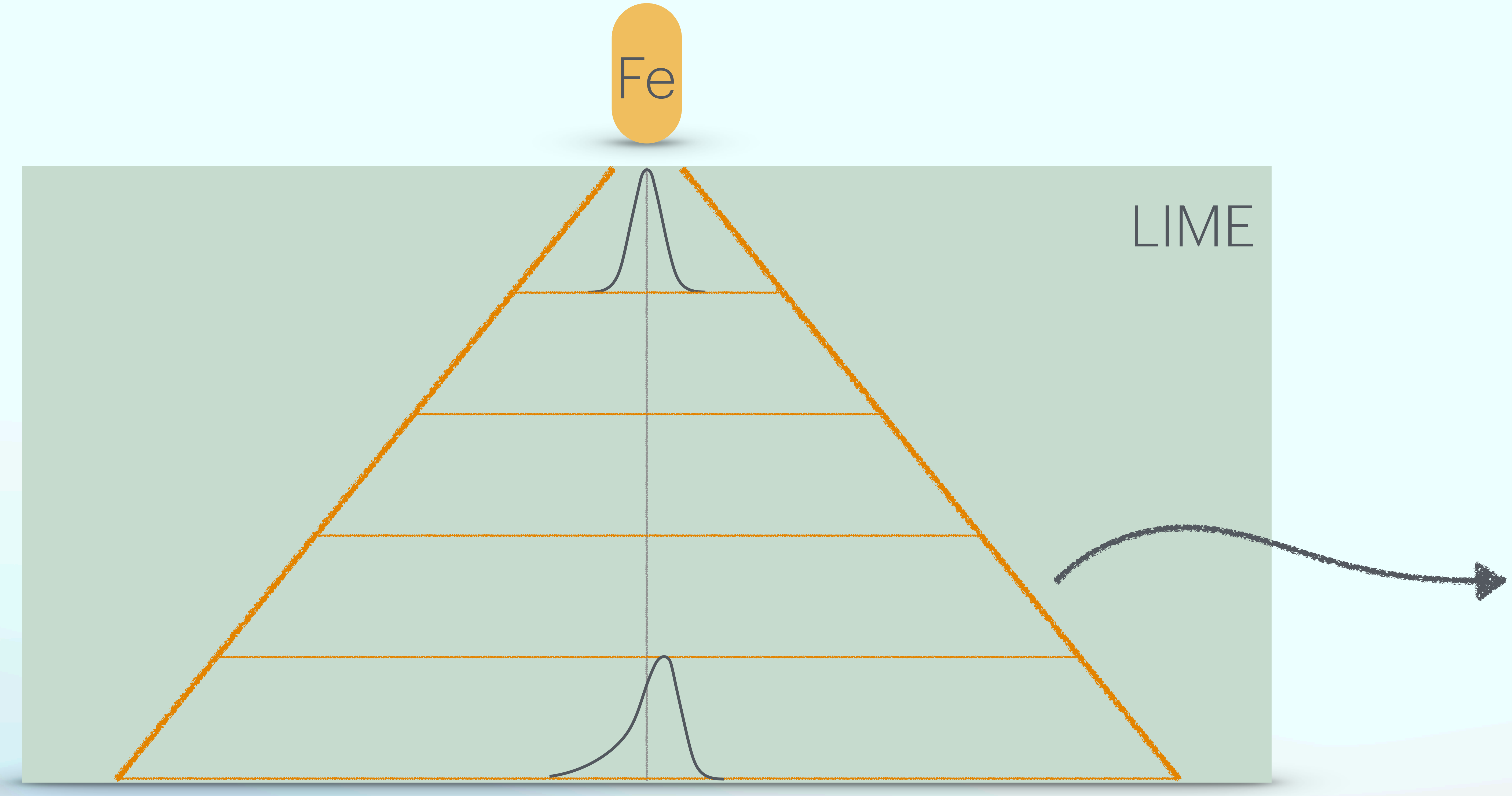
# Height scan - Observed data



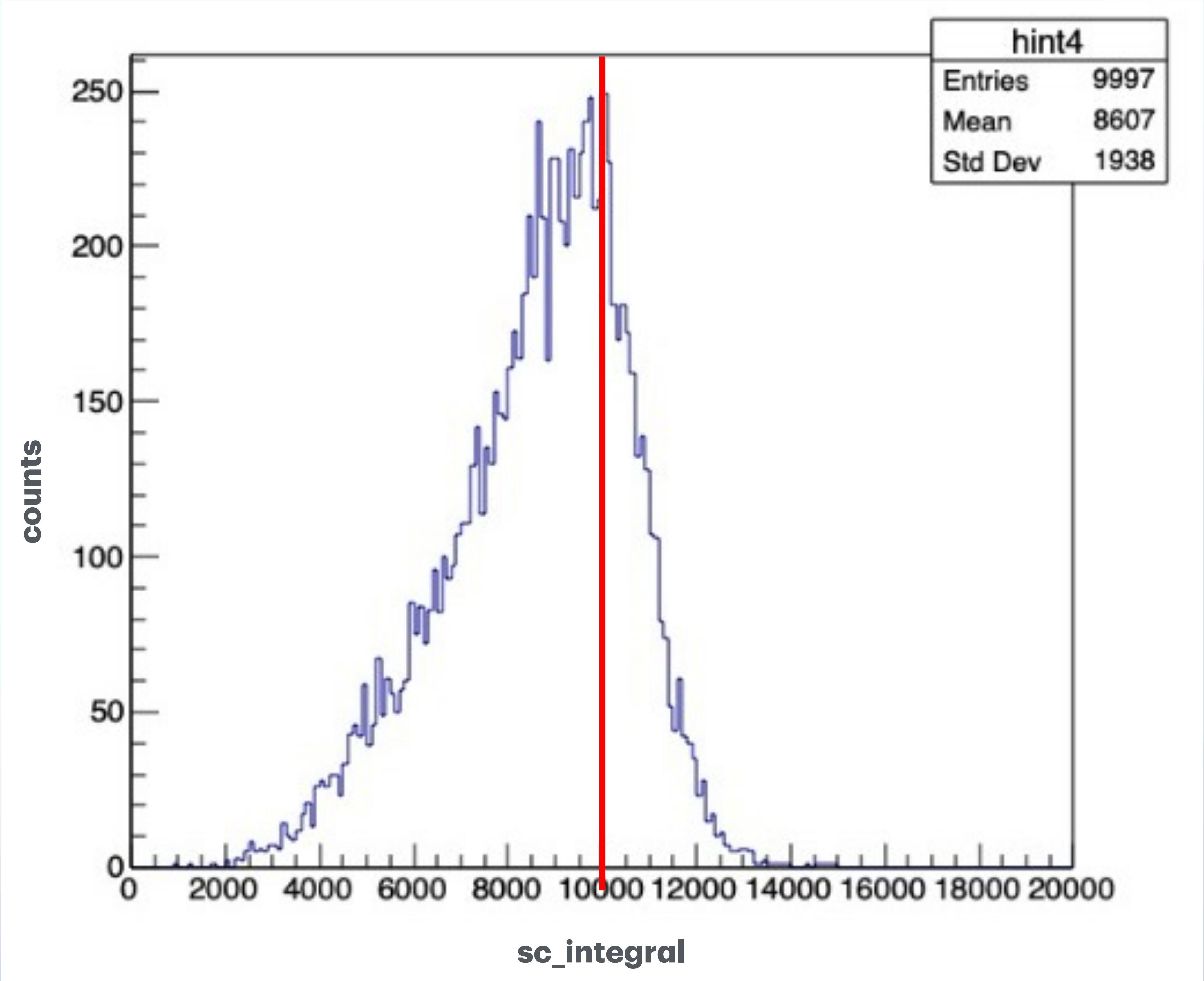
- Third slice



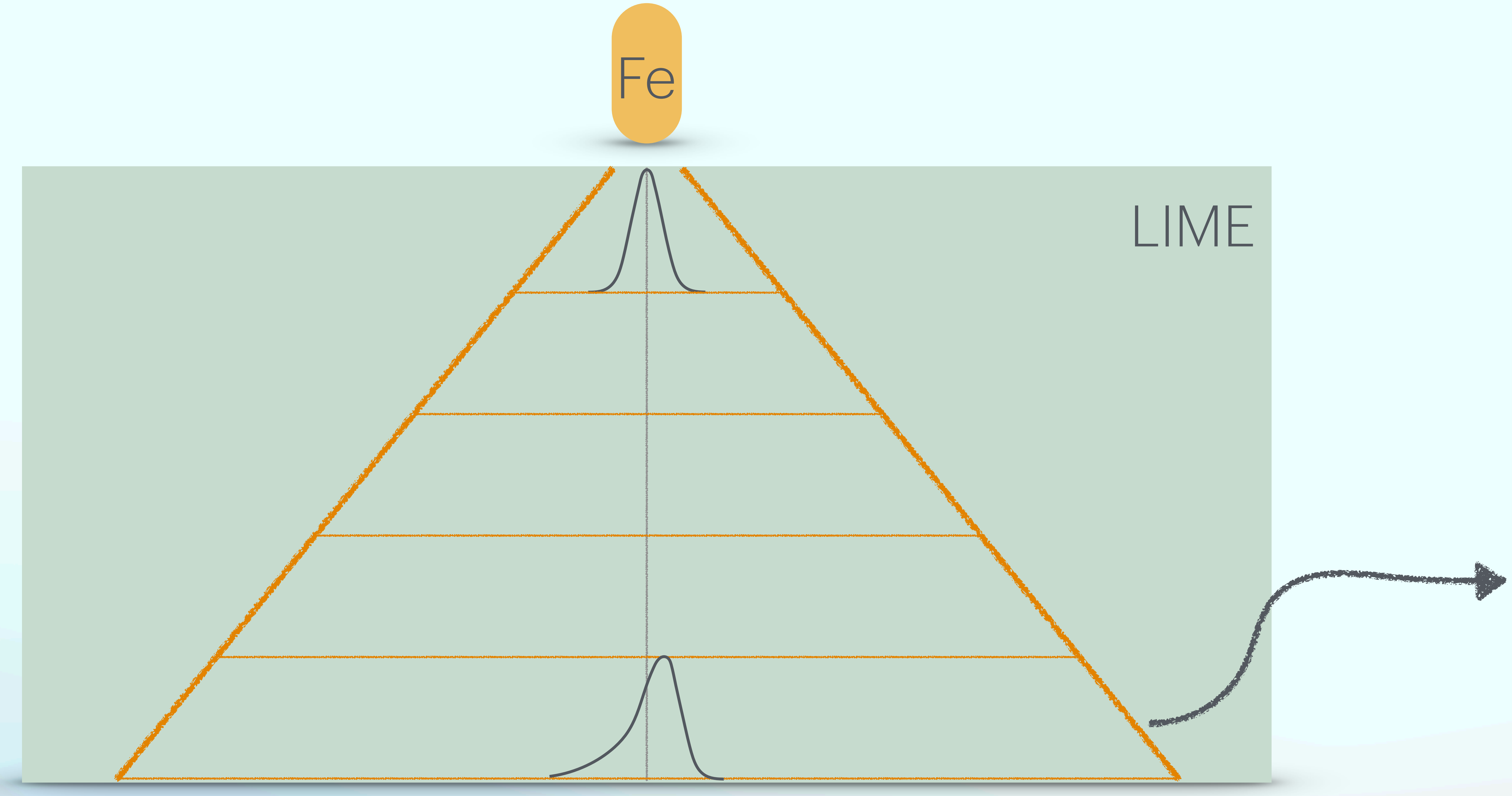
# Height scan - Observed data



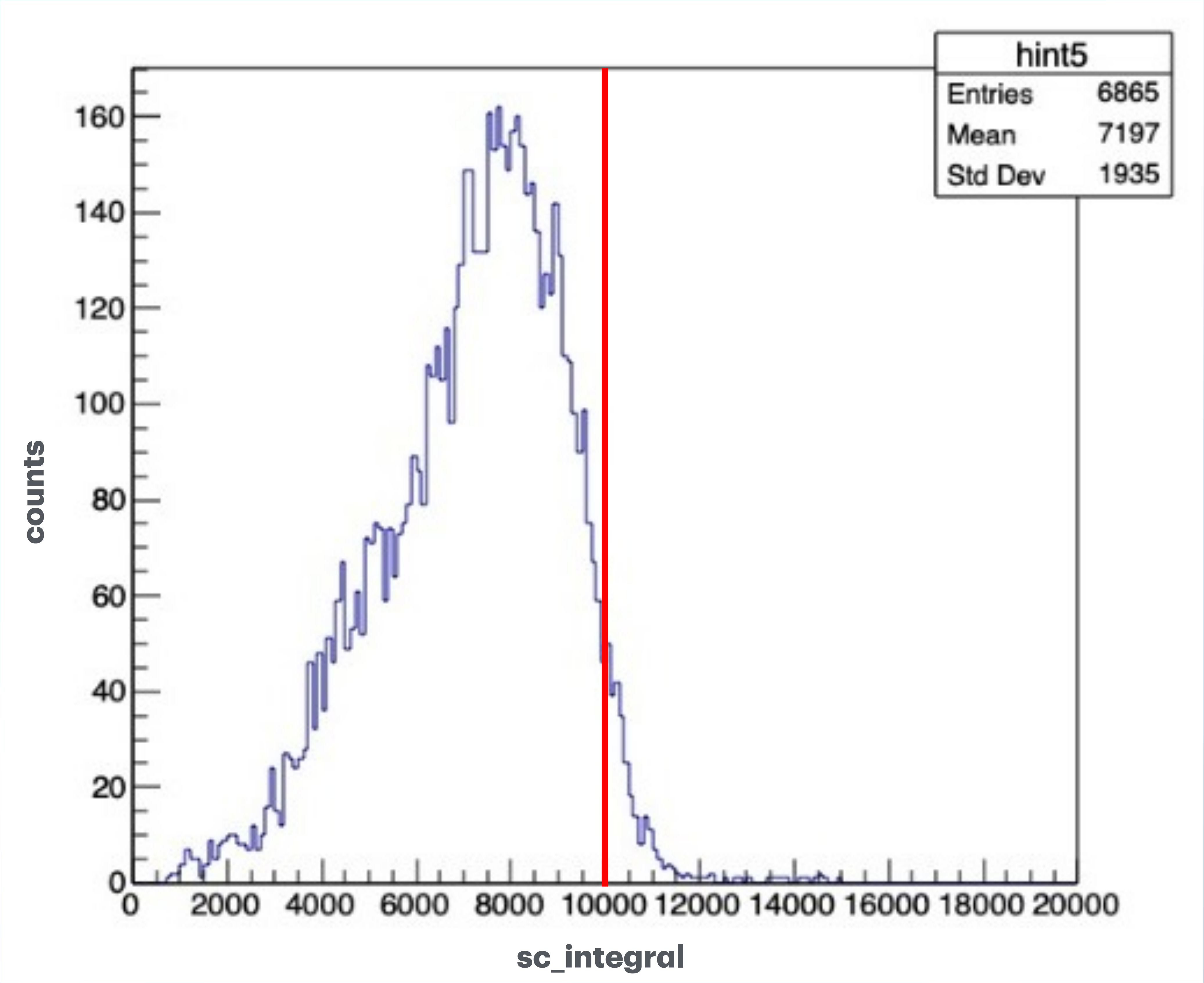
- Fourth slice



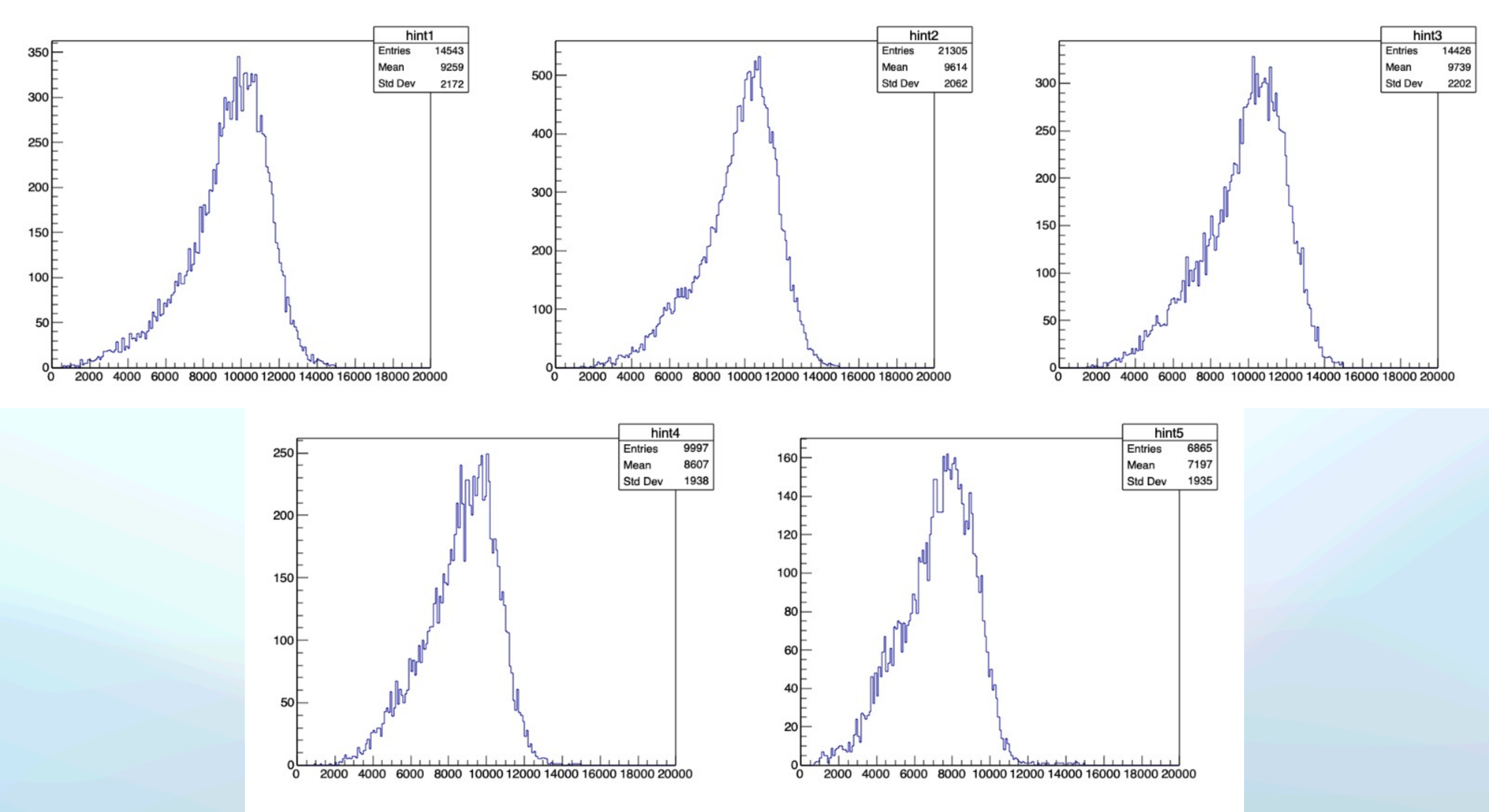
# Height scan - Observed data



- Fifth slice



# Height scan - Observed data (full view)

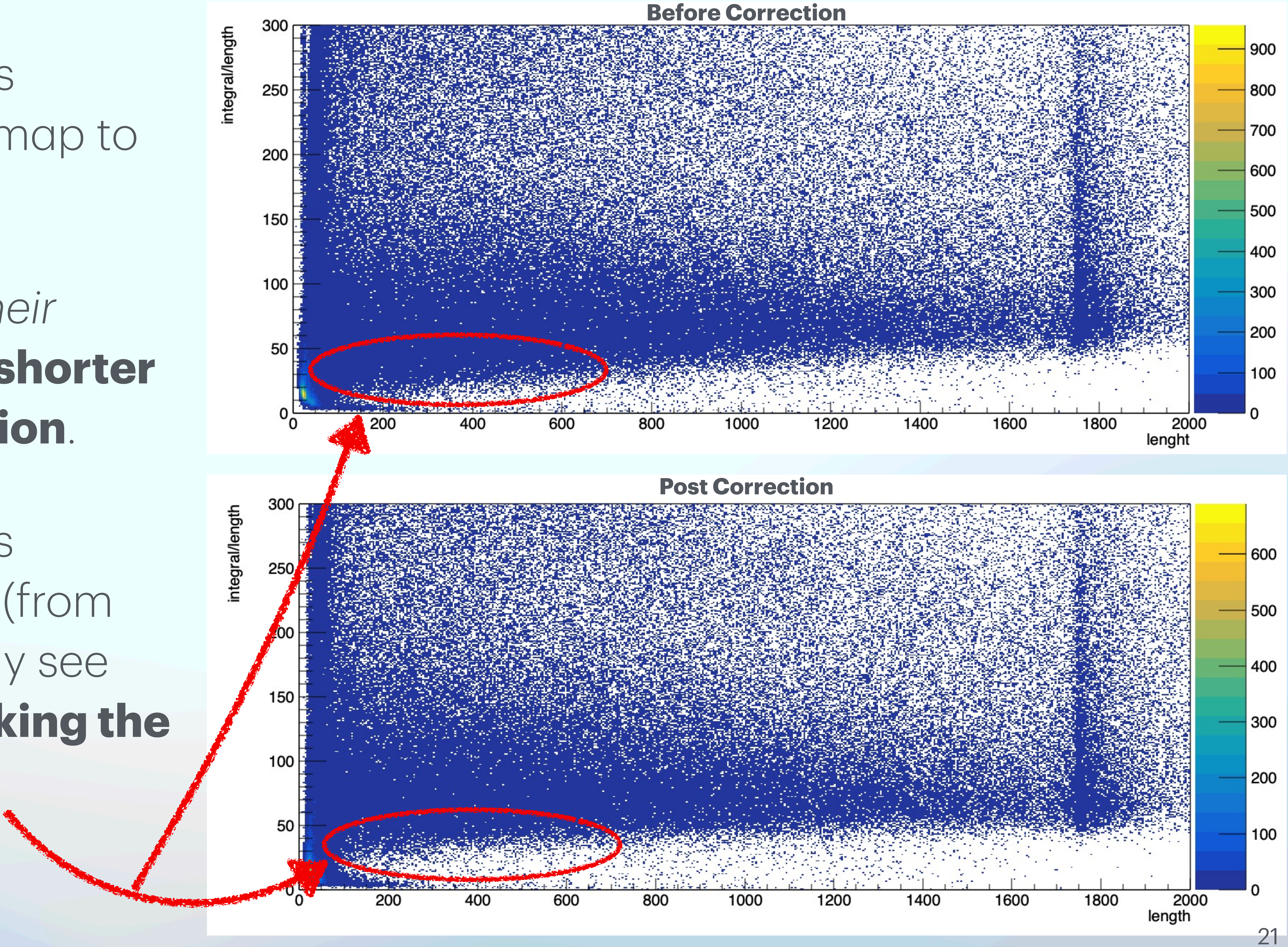


# Correction map consequences on the MIP band

- Let's now have a look at the results obtained applying the correction map to everything (also to MIPs).

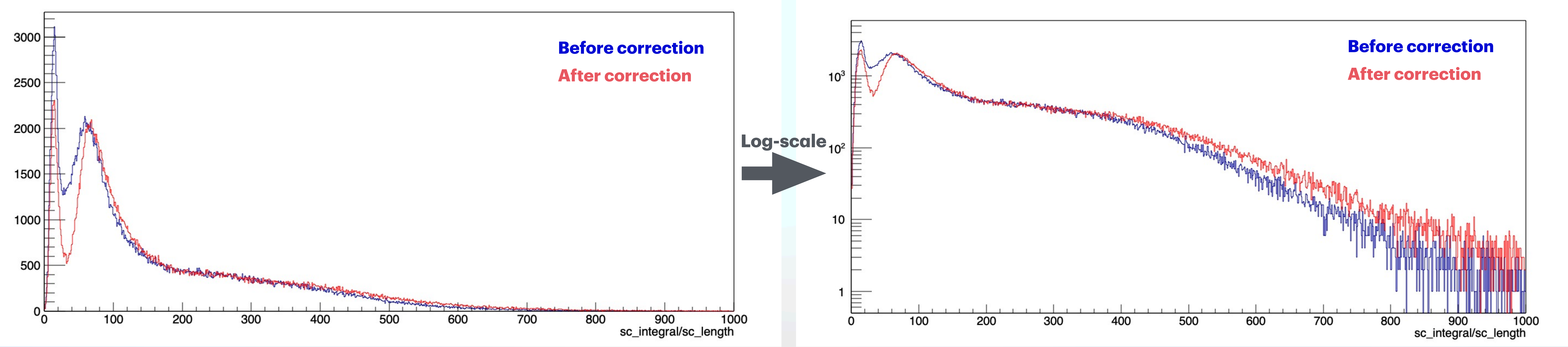
- In general, *longer tracks gain as their mean point*, while we expect that **shorter tracks needs a stronger correction**.

- Looking at sc_integral/sc_length vs sc_length pre and post correction (from *low-activity Eu runs*), we can clearly see how **the correction map is shrinking the low-length region!**



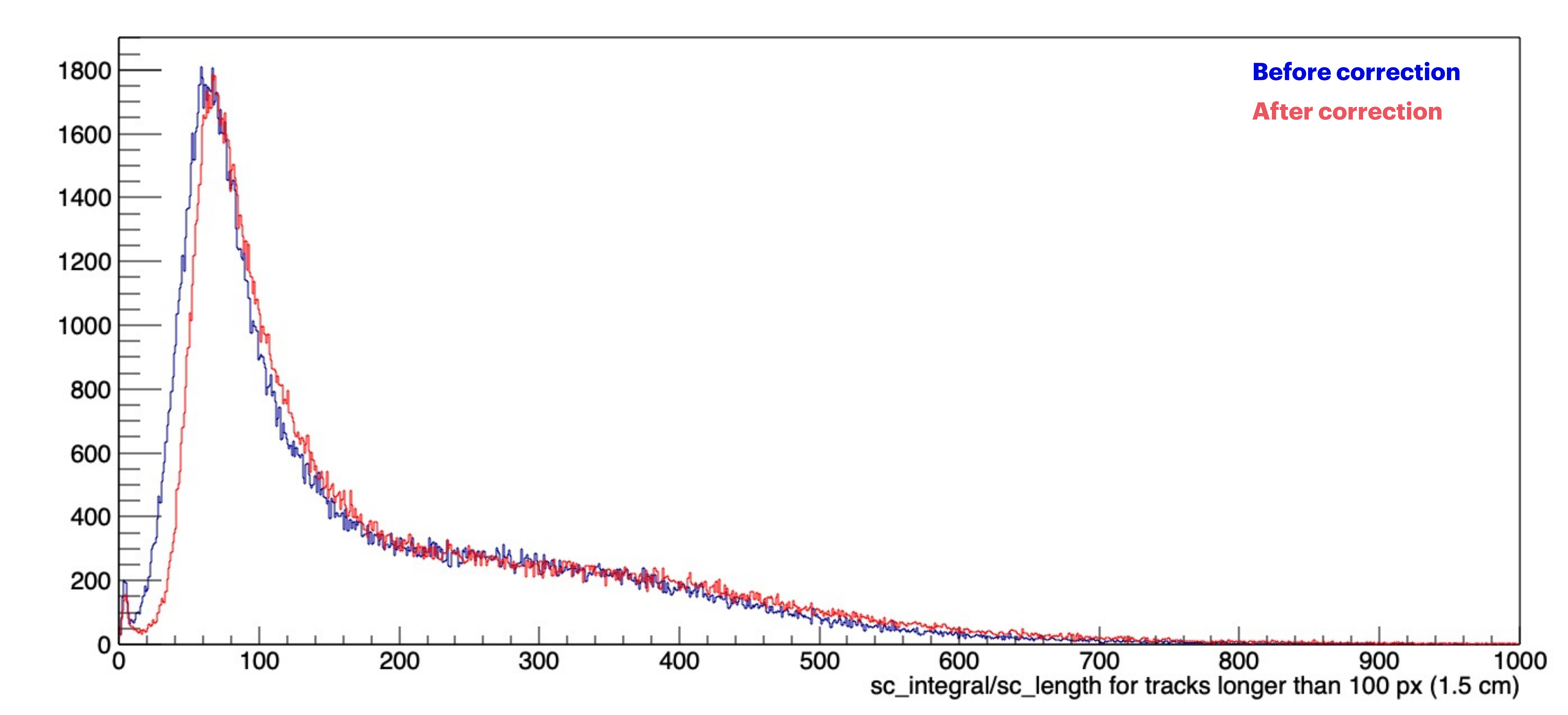
# Correction map consequences on the MIP band

- Looking at the sc_integral/sc_lenght we can see more in detail the MIP band shrinking **pre** and **post** correction.



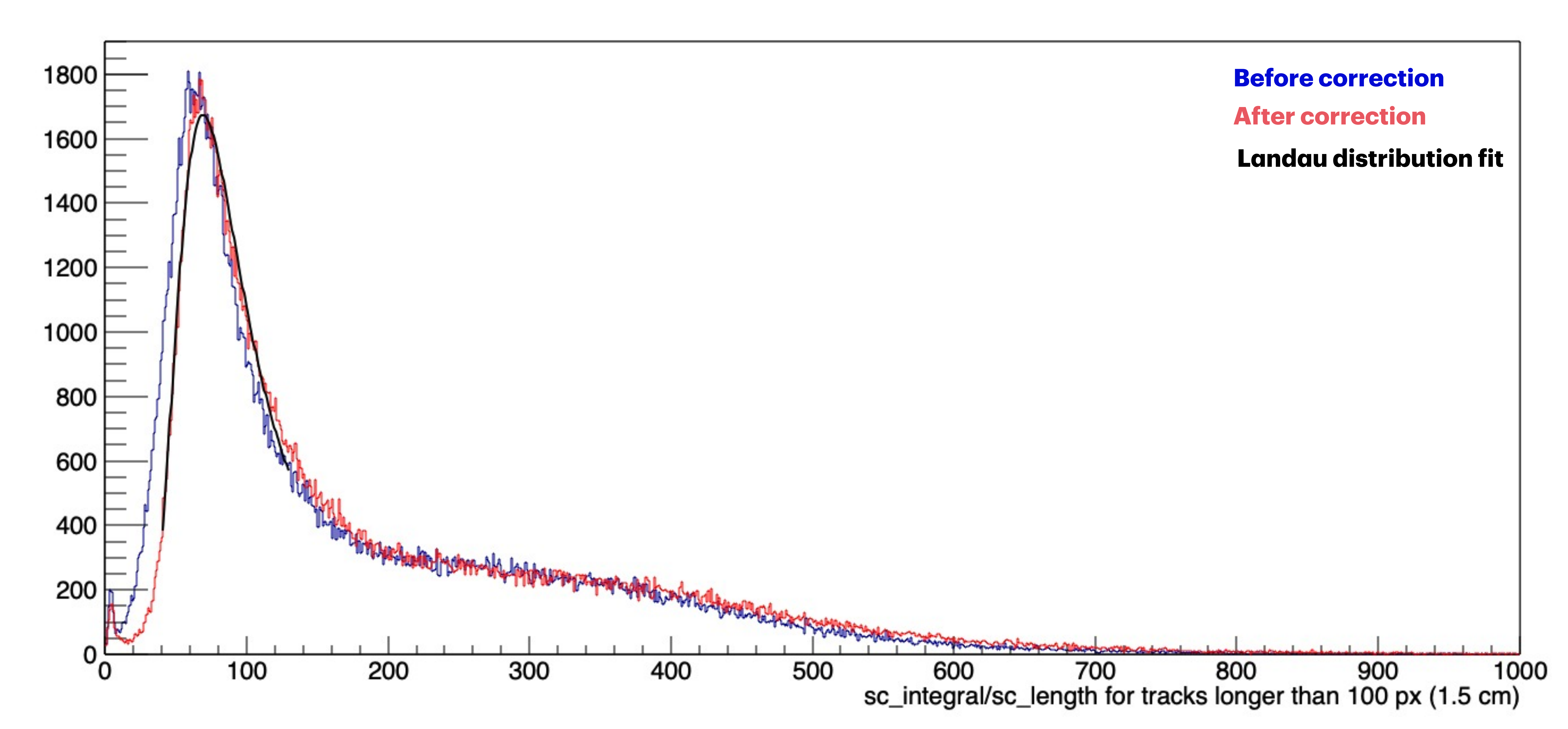
# Correction map consequences on the MIP band

- Moreover, we can exclude from the plot previous plot tracks shorter than 100 pixels (~1.5 cm).



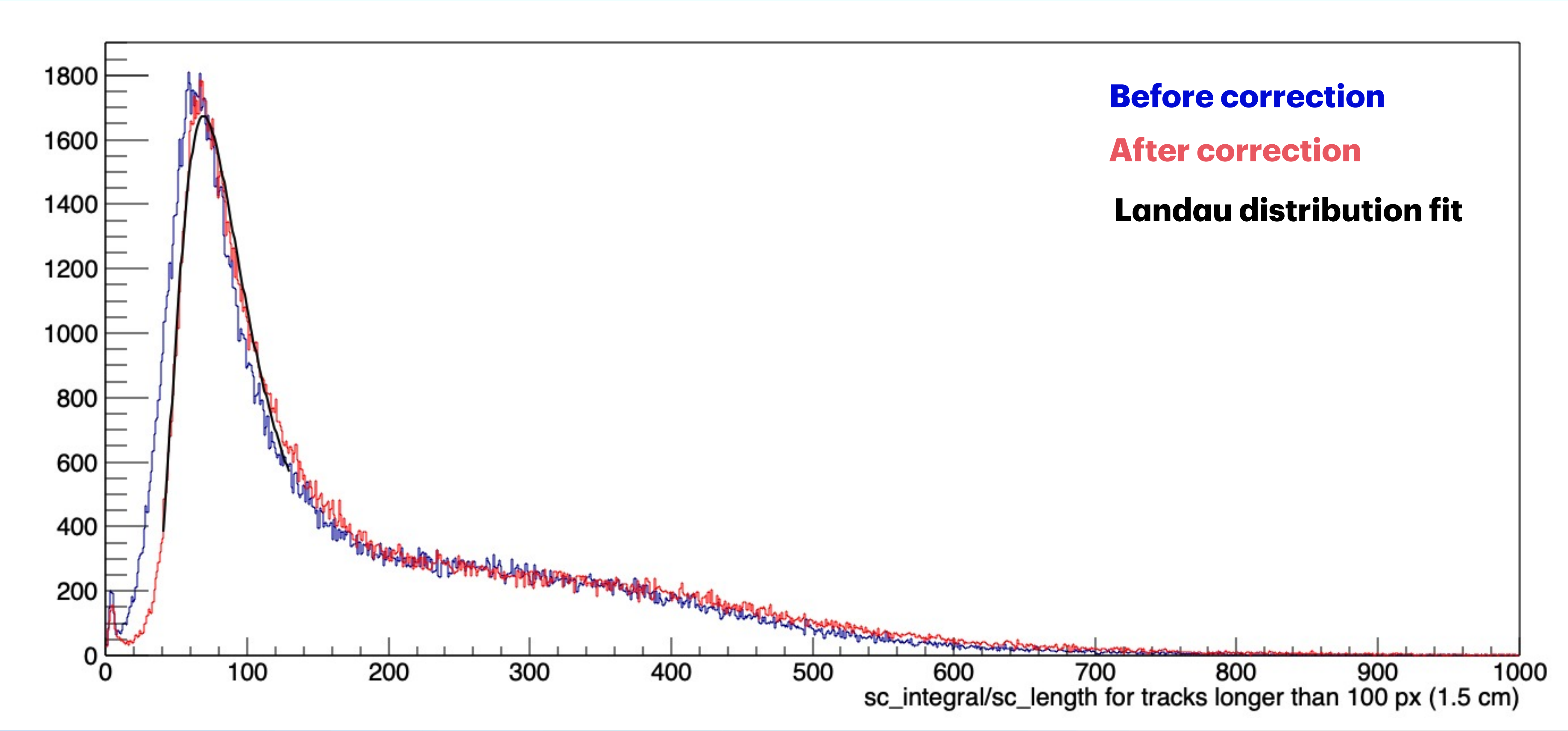
# Correction map consequences on the MIP band

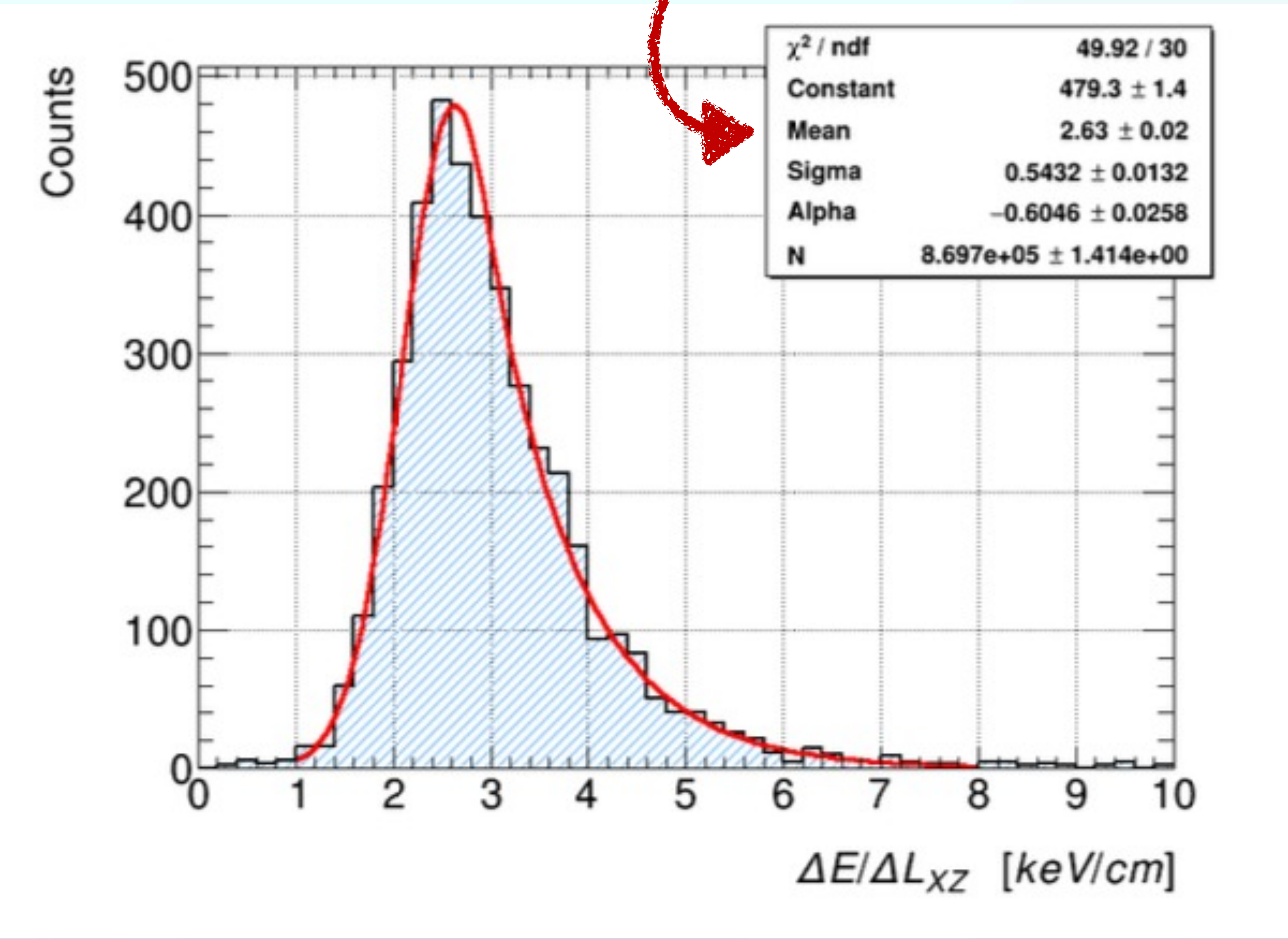
- We can fit the red curve with a **Landau distribution** which leads to a MPV of 73 counts/px.



# Correction map consequences on the MIP band

- If we calibrate the energy with the Iron peak with a rough 6000 eV : 10000 counts
- The MPV becomes **44 eV/0.15 mm = 290 eV/mm**, compatible with **Davide Marin measure**.





# Conclusions

- We saw that **the correction map works pretty well when correcting Fe data**. More in detail:

  ✓ It is retroactive.

  ✓ It *always* upgrades the resolution.

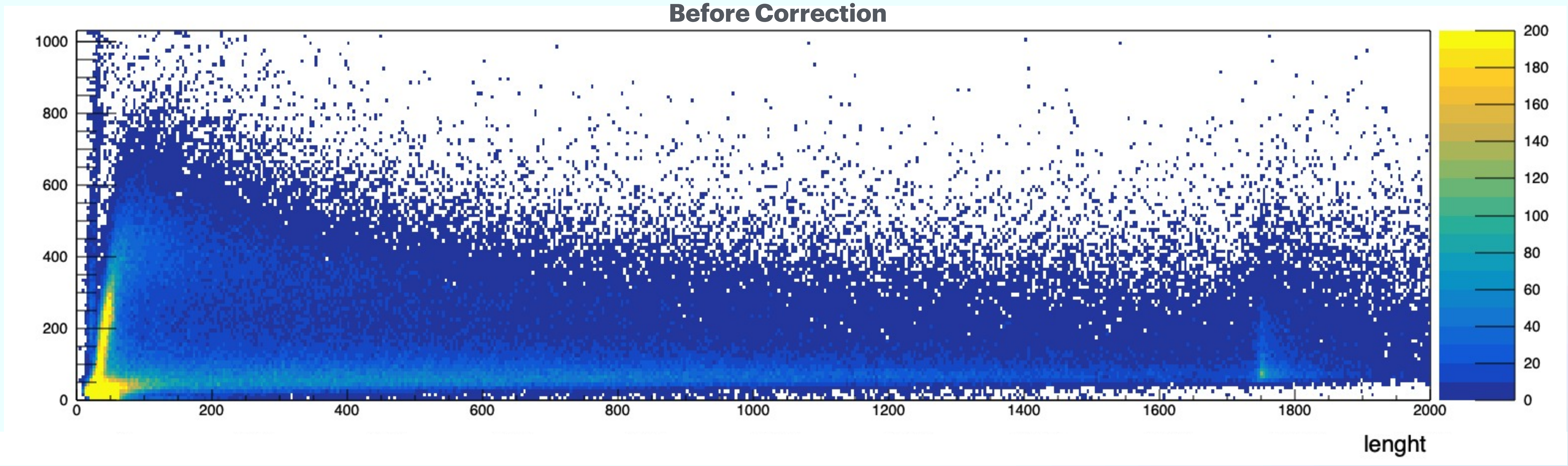
- **The intrinsic saturation effect**, which we explored with the height scan, **is clearly subdominant w.r.t. the observed disuniformity**. In fact we saw that the iron peak is shifted backwards (in terms of sc_integral) in the lower regions of the GEM plane w.r.t. the upper regions, contrary to what we expected.

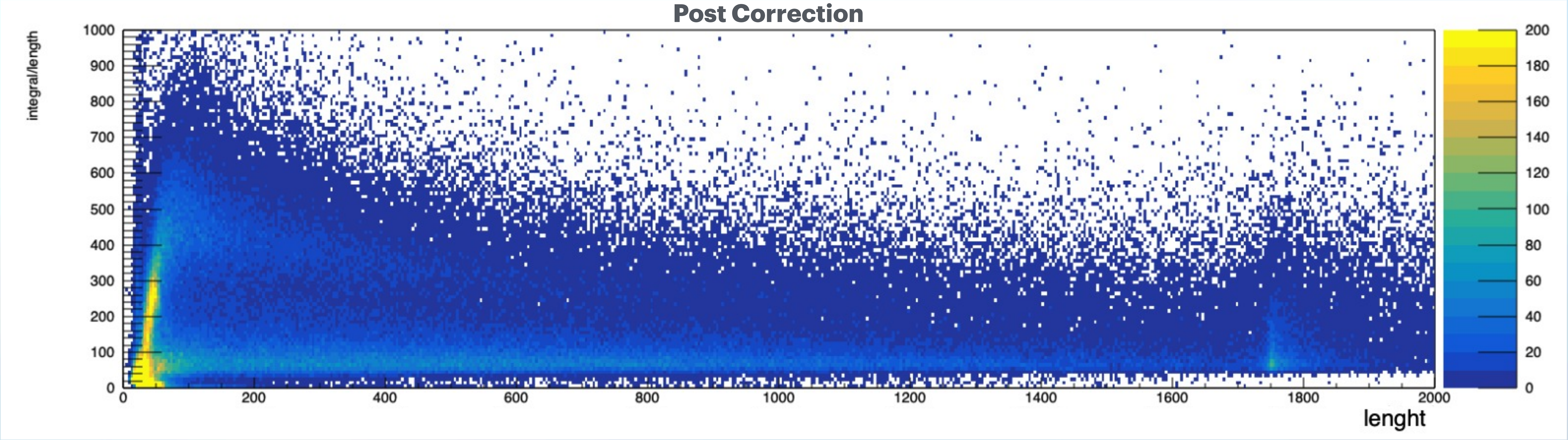
- We also saw that **the benefits appears to extend also to the MIP band**.

- More test are foreseen to cross-check the map behaviour in different situations.

# Correction map consequences on the MIP band

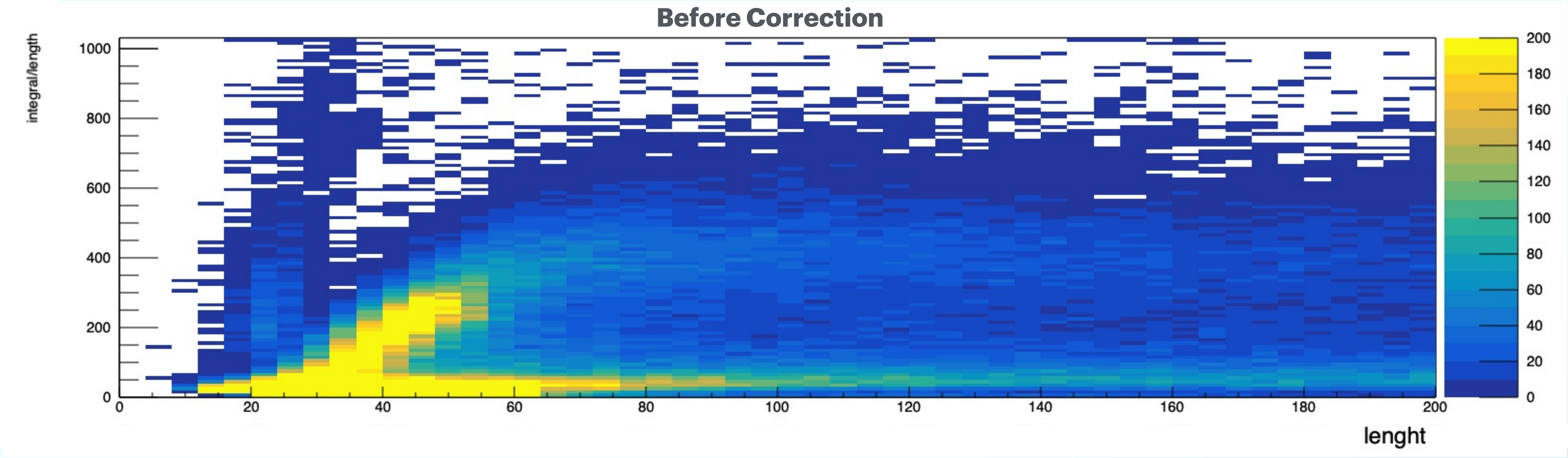
- Zooming **out** the y-axis from slide 21 plots.

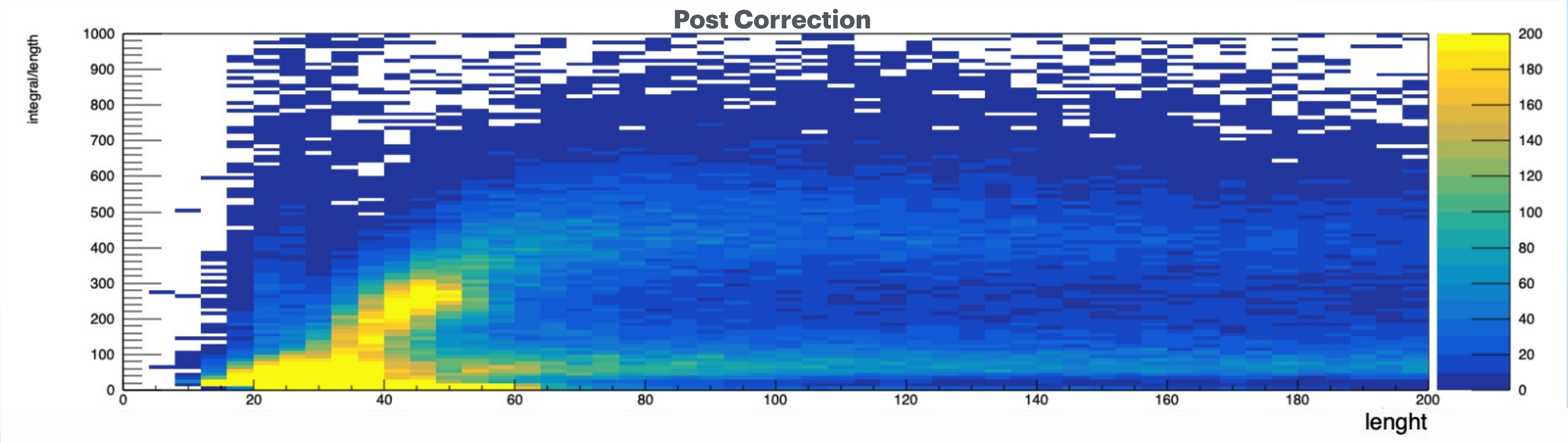




# Correction map consequences on the MIP band

- Zooming **in** the y-axis from slide 21 plots.


Before Correction
integral/length
lenght


Post Correction
integral/length
lenght