Impact of filtering on Cygno’s images

Guilherme Lopes, Igor Abritta, Rafael Nóbrega

Motivation

- How to evaluate the performance of preprocessing algorithms for CYGNO?
- What is the impact of different preprocessing algorithms on energy estimation?
- Is it possible to get same clustering results decreasing the number of points sent to dbscan?

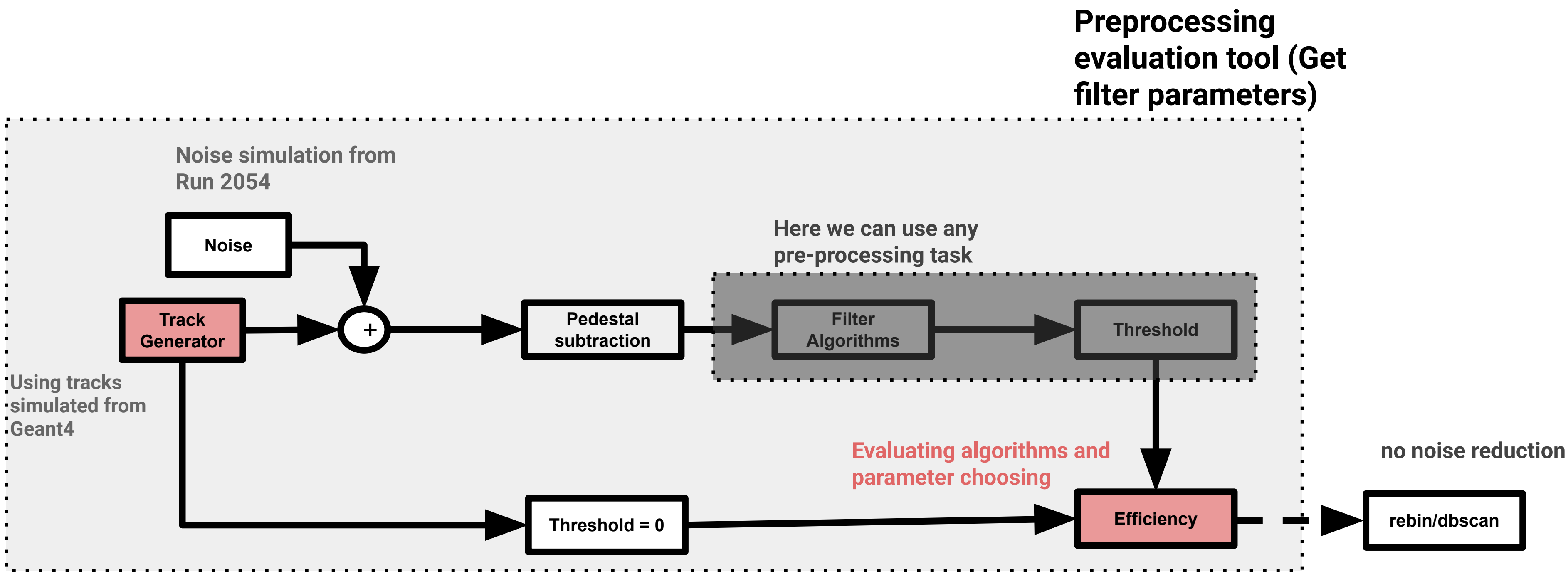
Objective

- Find a proper methodology to assess the performance of preprocessing algorithms for CYGNO → propose a test environment with this end
- Evaluate the impact of some filters on efficiency/false-alarm and energy estimation using simulated data
- **Evaluate the impact of some filters using real data.**

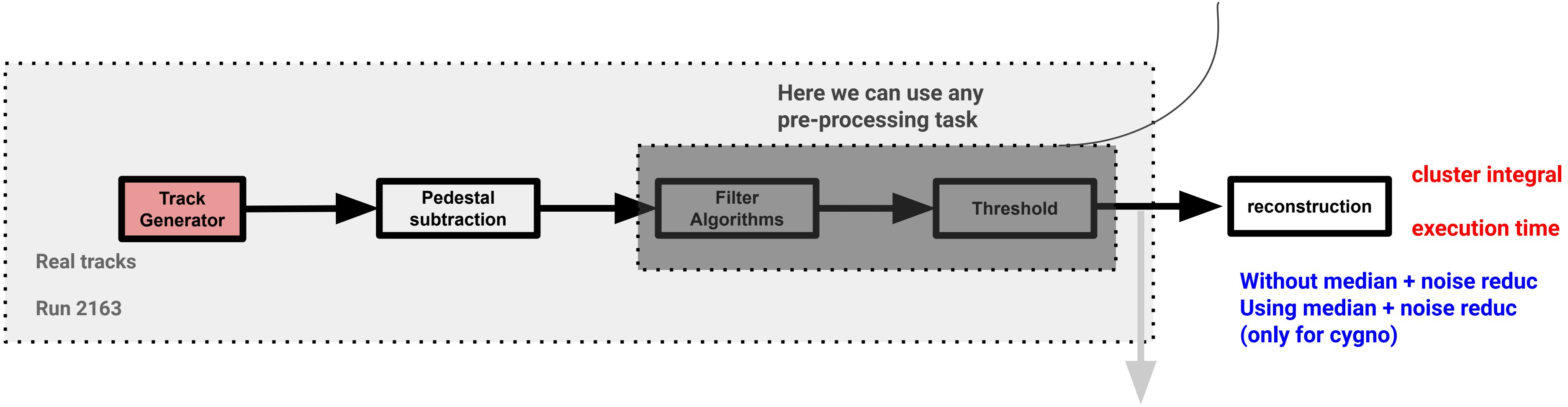
https://agenda.infn.it/event/25064/contributions/127002/attachments/77918/100522/Presentation.pdf

Flowchart overview

Simulation tool flowchart


Preprocessing evaluation tool (Get filter parameters)
Noise simulation from Run 2054
Noise
Track Generator
+
Pedestal subtraction
Here we can use any pre-processing task
Filter Algorithms
Threshold
Using tracks simulated from Geant4
Evaluating algorithms and parameter choosing
no noise reduction
Threshold = 0
Efficiency
rebin/dbscan

Evaluating using real data



Parameters defined by simulation

Here we can use any pre-processing task

Track Generator

Pedestal subtraction

Filter Algorithms

Threshold

reconstruction

Real tracks

Run 2163

cluster integral

execution time

Without median + noise reduc

Using median + noise reduc

(only for cygno)

Number of points after threshold

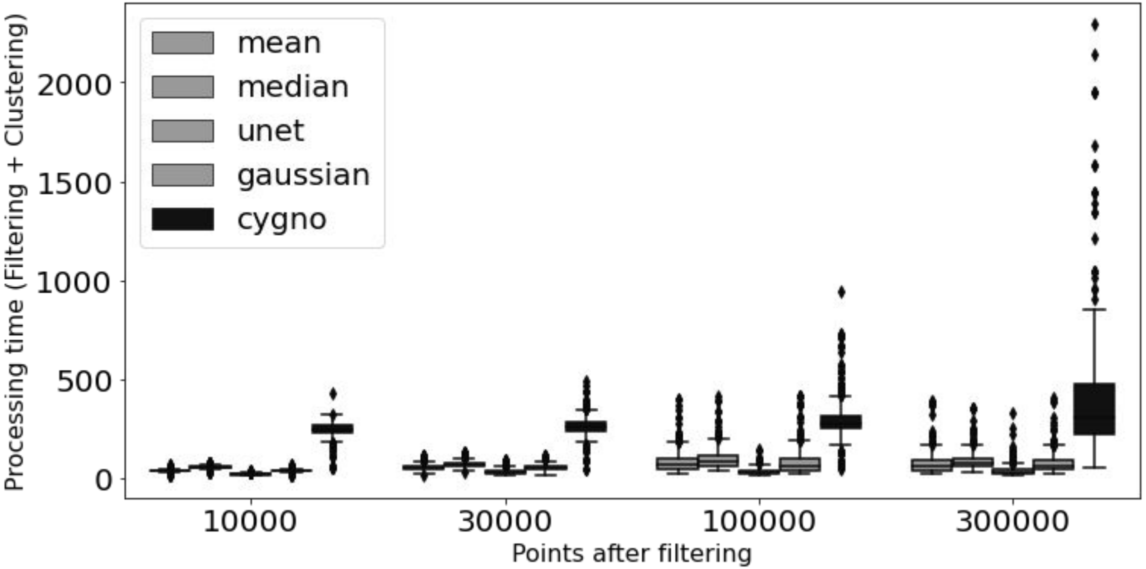
Setup

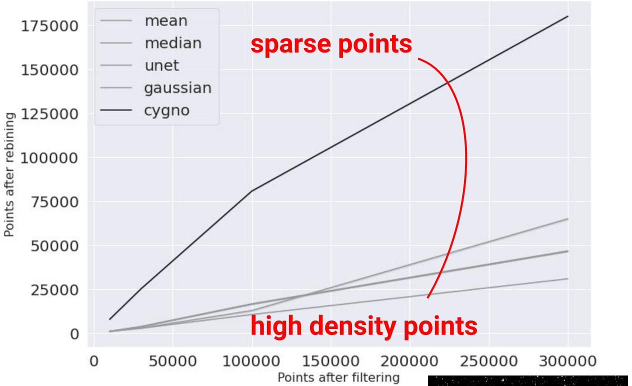
- ***Run:***
  - ***2163;***
- ***Filters:***
  - ***cygno (n*std);***
  - ***median (w = 17);***
  - ***u-net (Trained using simulation data);***
  - ***mean (w = 15);***
  - ***gaussian (w = 15).***
- ***Points after threshold (N):***
  - ***30000;***
  - ***100000;***
  - ***300000 (n ~ 1.3).***
- ***Median + noise reduction (cygno)***
- ***dbscan: 3D, iterative=4, eps=[1, 2.5, 5.8 4]   min_samples=[1, 420 30, 20]***

Using median+noise red (cygno)

Using median + noise reduction (cygno)

Execution time



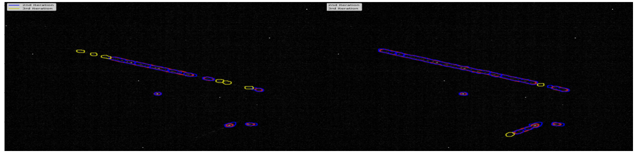


sparse points

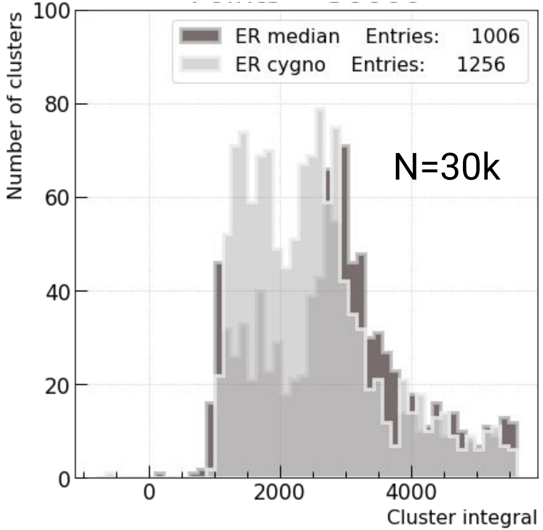
high density points

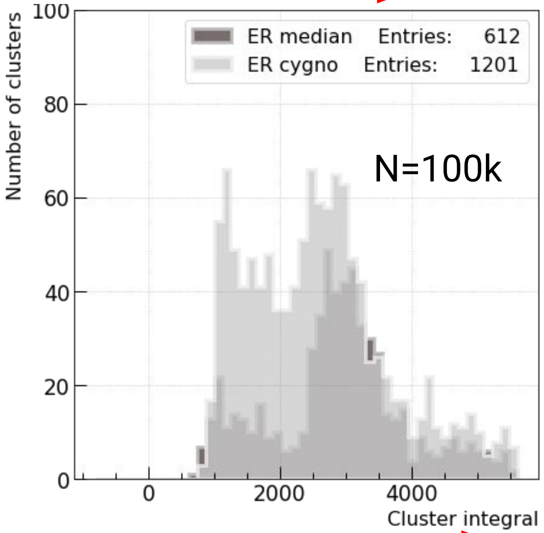
Using median + noise reduction (cygno)

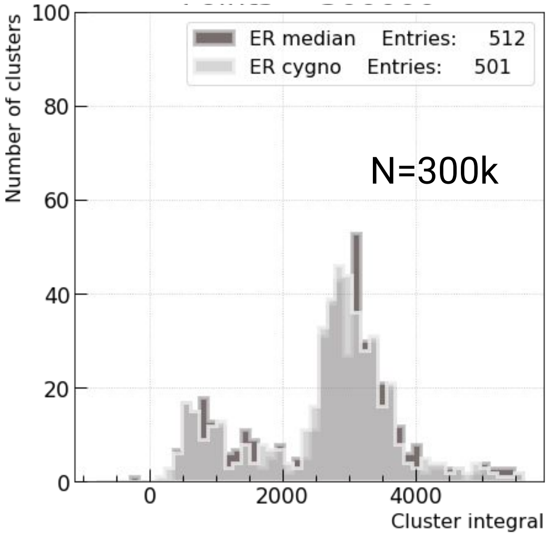
Cluster integral: median x cygno, N same



Increasing N





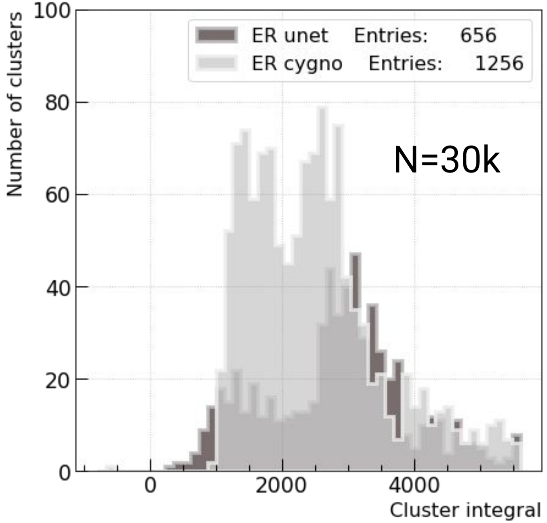


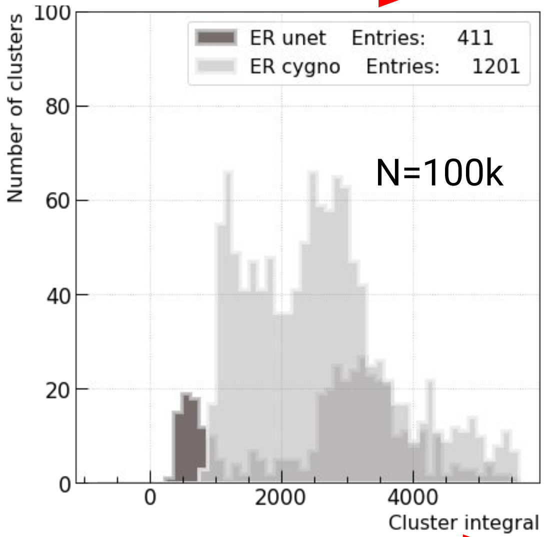
Decreasing #clusters until some N value

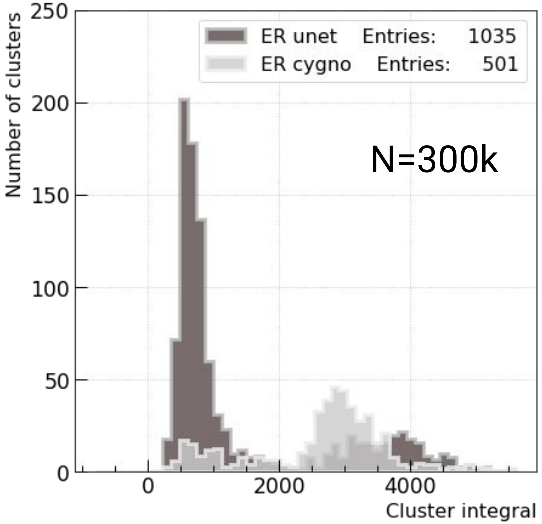
Using median + noise reduction (cygno)

Cluster integral: unet x cygno, N same

Increasing N





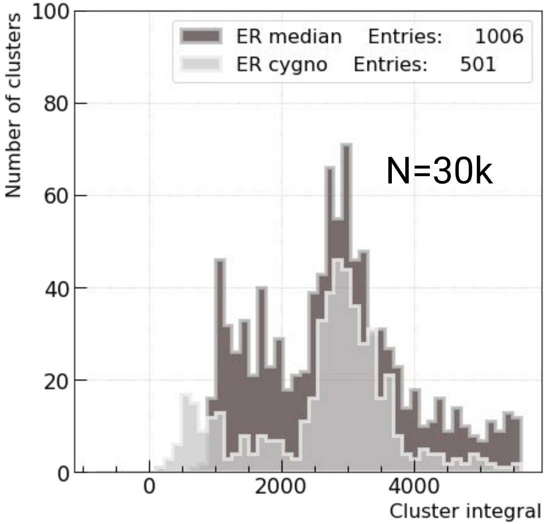


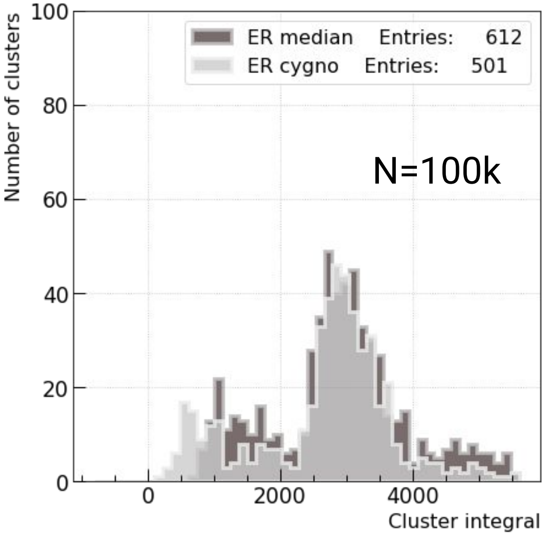
Decreasing #clusters until some N value

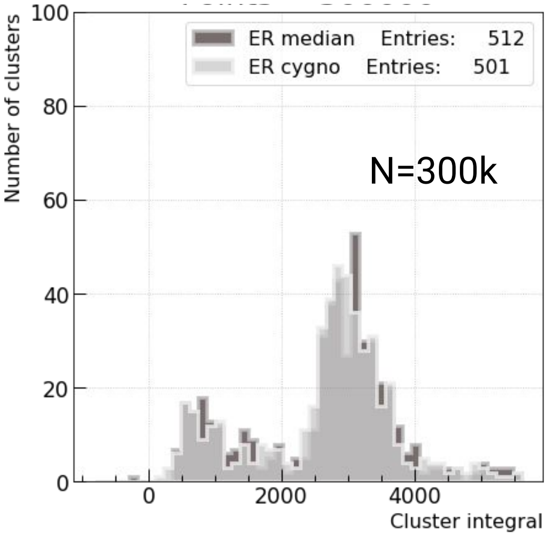
Using median + noise reduction (cygno)

Cluster integral: median x cygno, keep N cygno = 300k

Increasing N





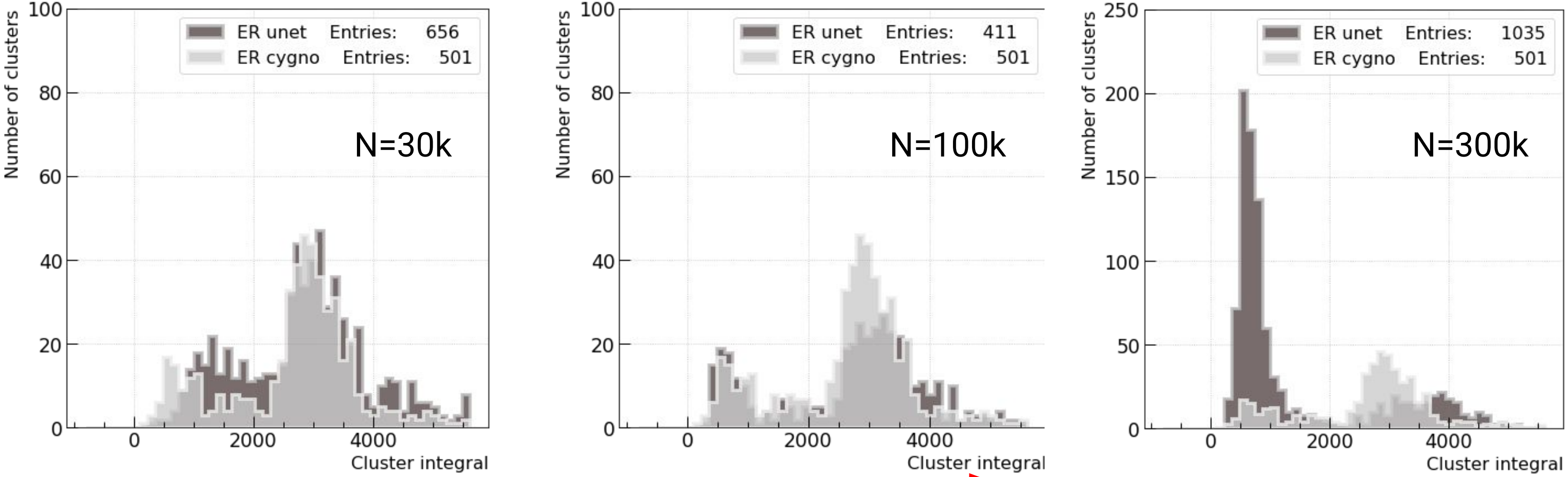


Decreasing #clusters until some N value

Using median + noise reduction (cygno)

Cluster integral: unet x cygno, keep N cygno = 300k

Increasing N

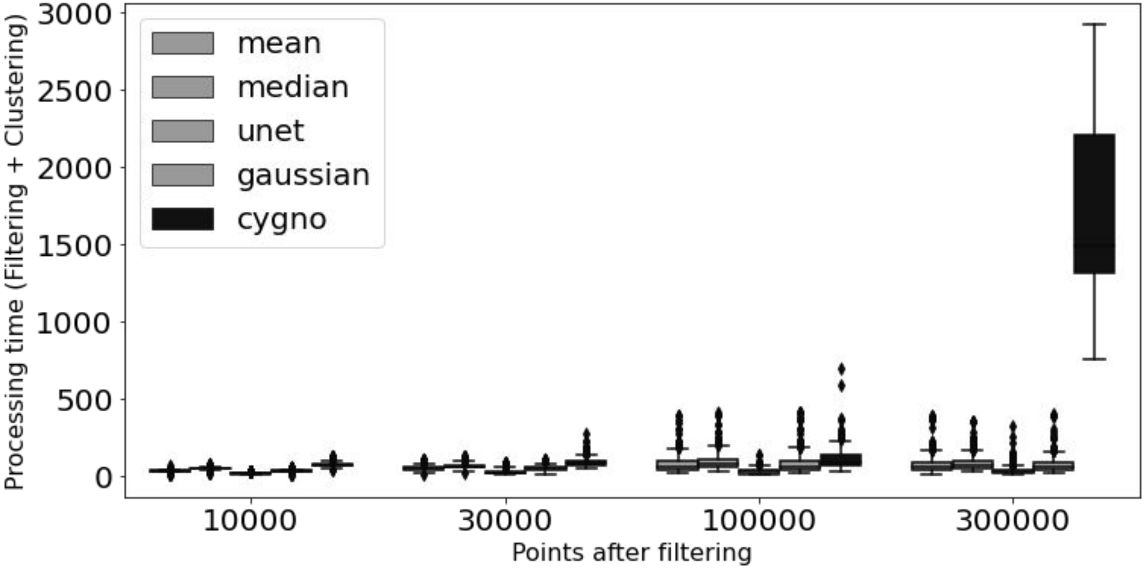


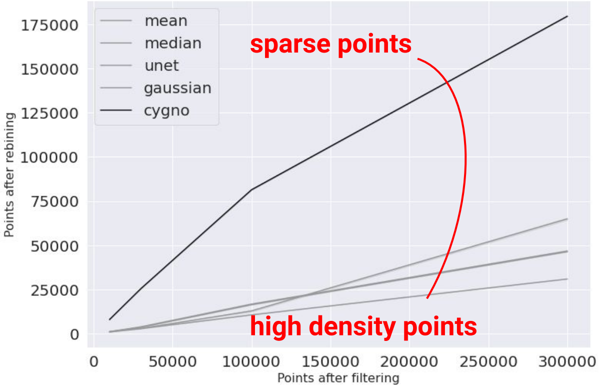
Decreasing #clusters until some N value

Without median+noise red

Without median + noise reduction

Execution time





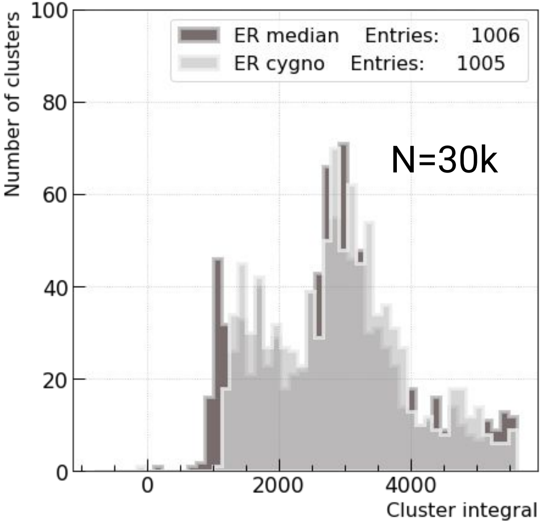
sparse points

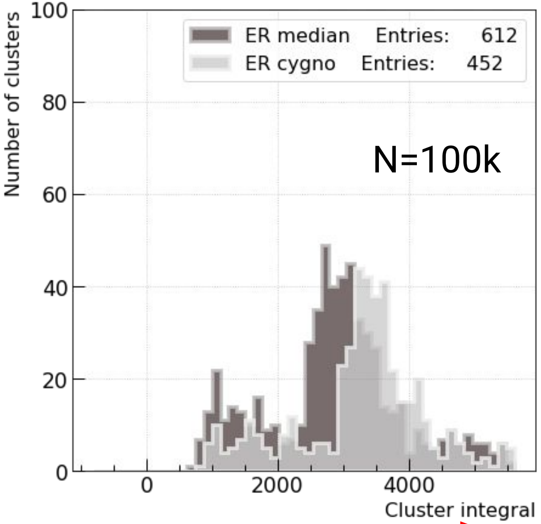
high density points

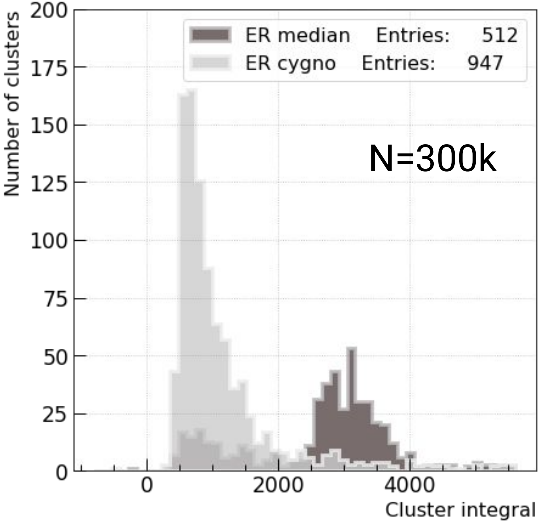
Without median + noise reduction

Cluster integral: cygno x median

Increasing N





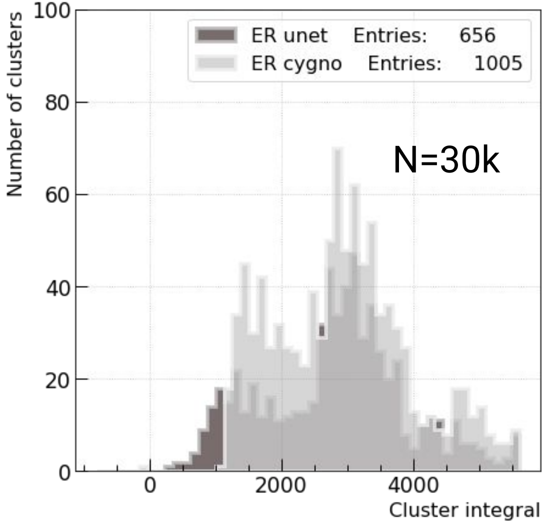


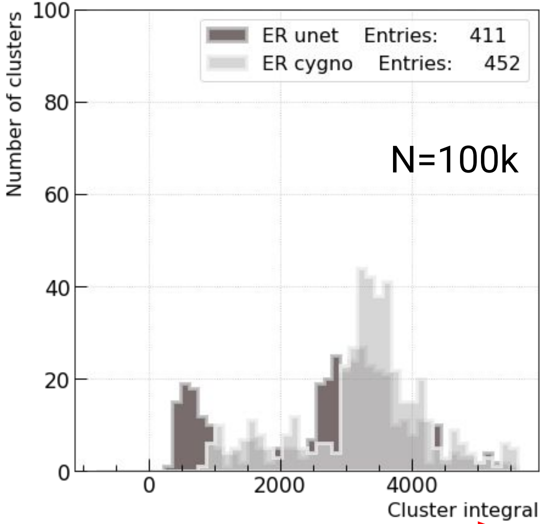
Decreasing #clusters until some N value

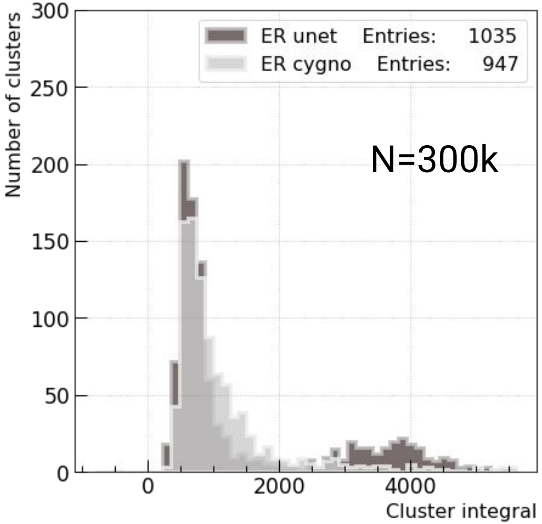
Without median + noise reduction

Cluster integral: cygno x unet

Increasing N



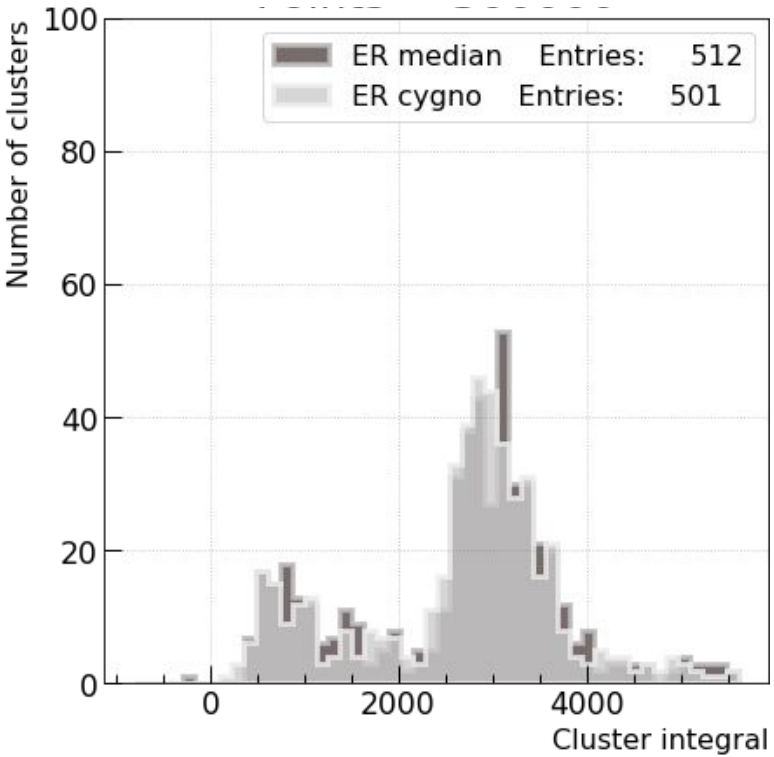




Decreasing #clusters until some N value

Without median + noise reduction

Best case (for now)



Algorithm	Total time (869)(1 core)
cygno	55h 30m
median	**12h 04m**

~ 4.5x faster

Conclusions

- ***Filters can improve the processing time of the reconstruction algorithm maintaining the energy distribution similar to the output of the algorithm used by the collaboration;***
- ***The improvements obtained are due to the removal of the noise reduction algorithm and the arrangement of pixels in the image after the filtering process;***
- ***Clustering parameters have been dimensioned for the cygno algorithm, adjusting this for the filters, we can obtain improvements.***