

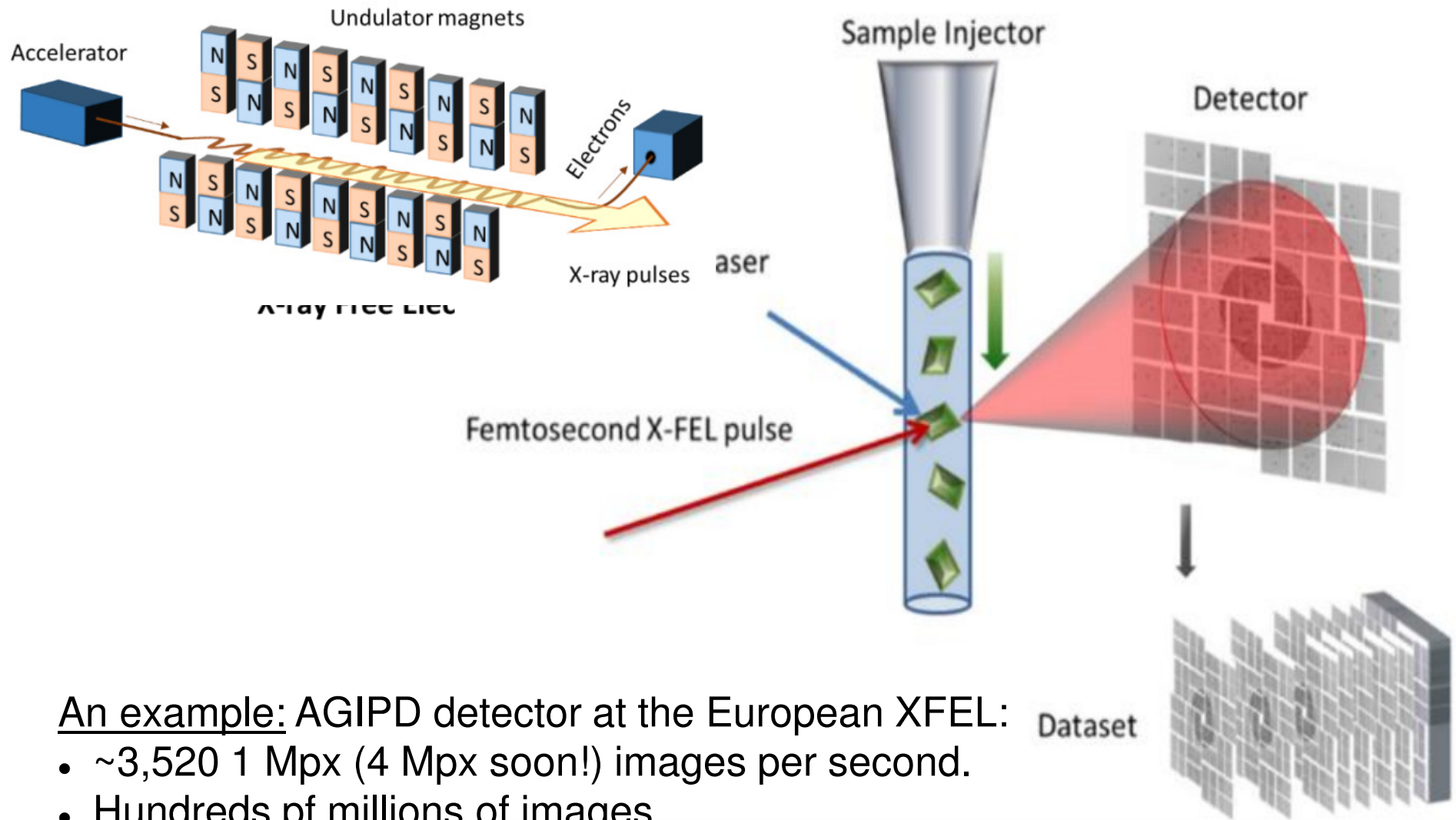
# Robust statistics for “Bad pixel mask”

Using Robust Statistics and Machine Learning methods to help with data reduction in FEL data analysis pipeline

Alireza Sadri  
Hamburg, 08.06.2020



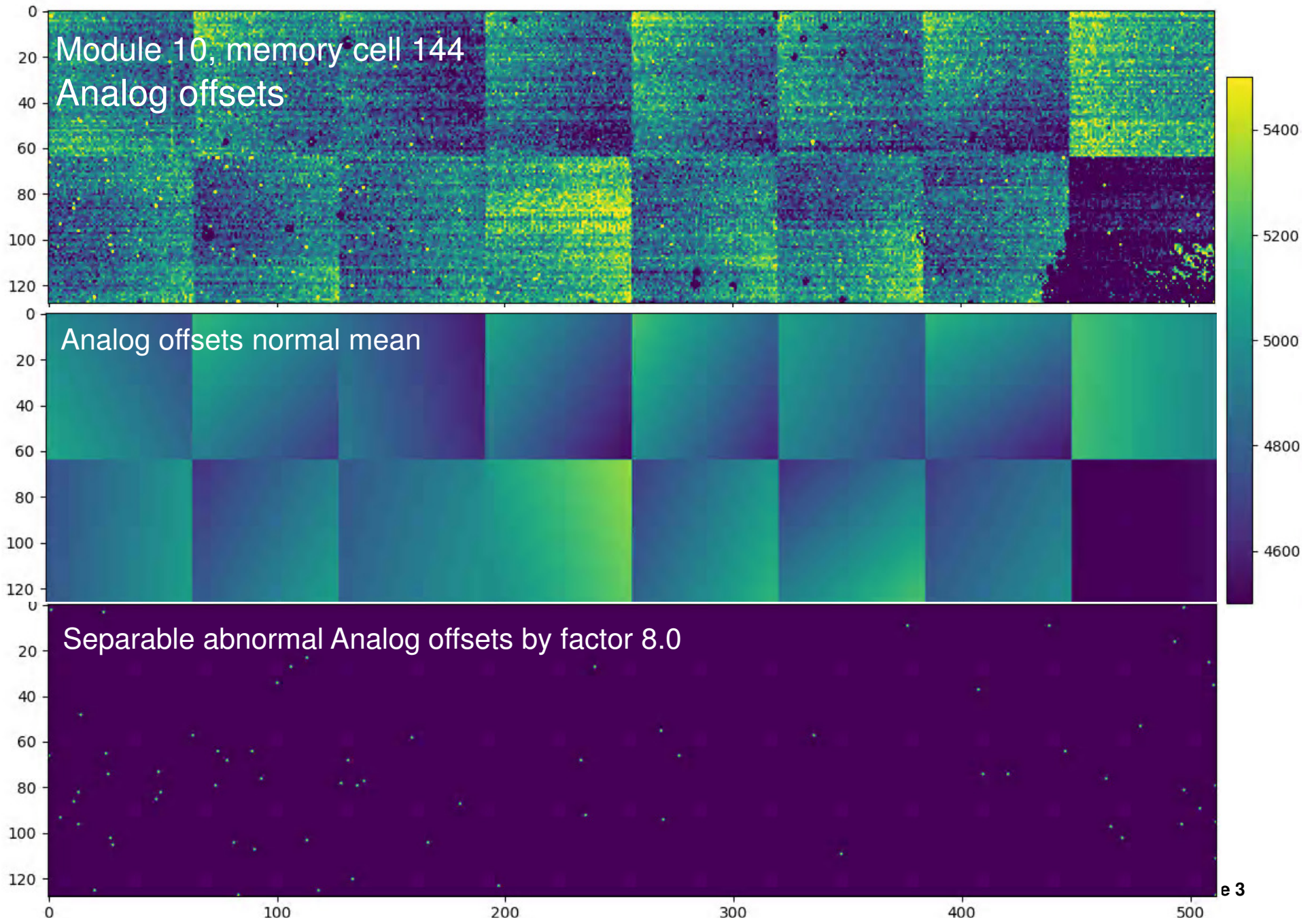
# FEL Data volume challenge



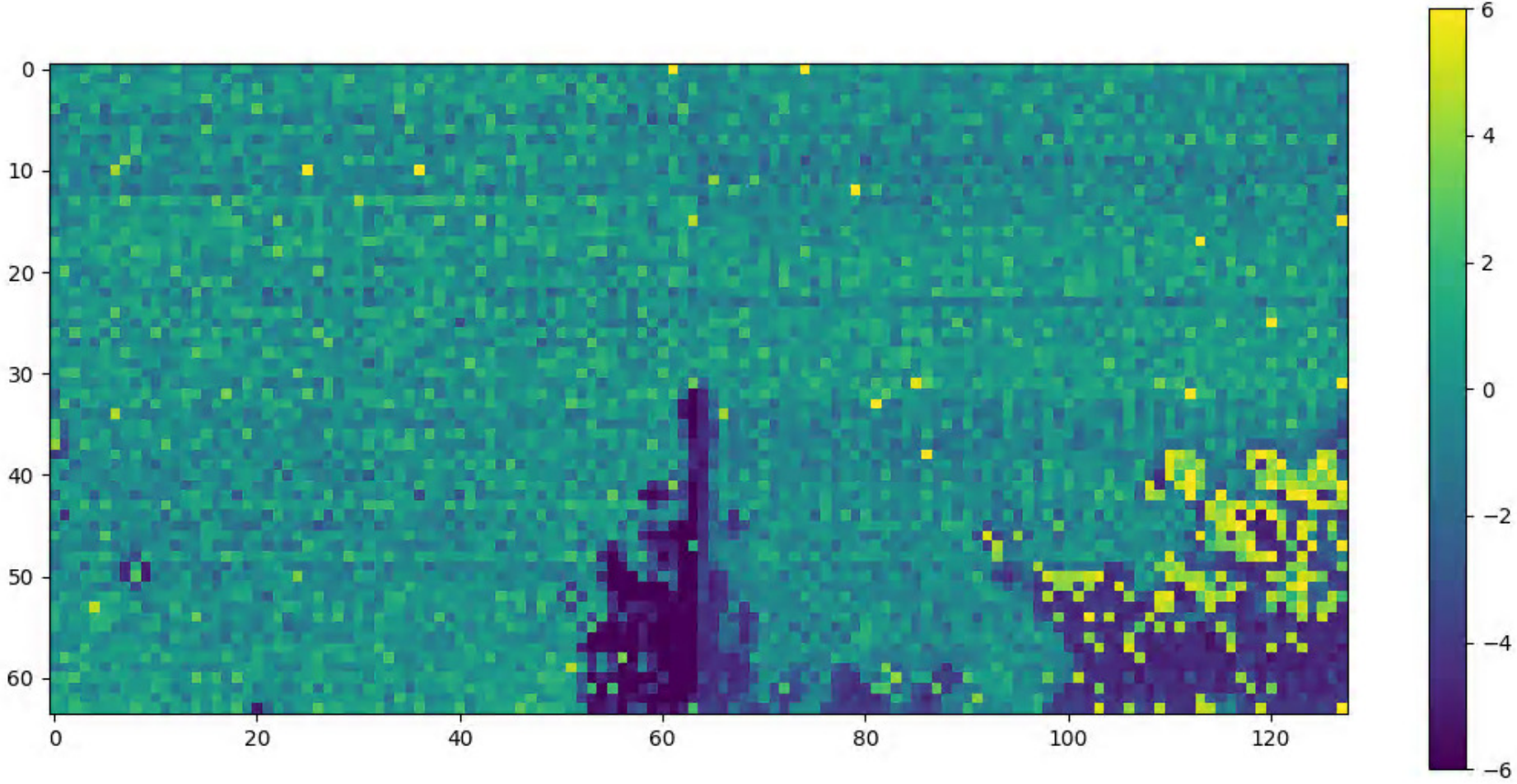
An example: AGIPD detector at the European XFEL:

- ~3,520 1 Mpx (4 Mpx soon!) images per second.
- Hundreds of millions of images
- We need maybe thousands of good images...

# Example: analog offsets



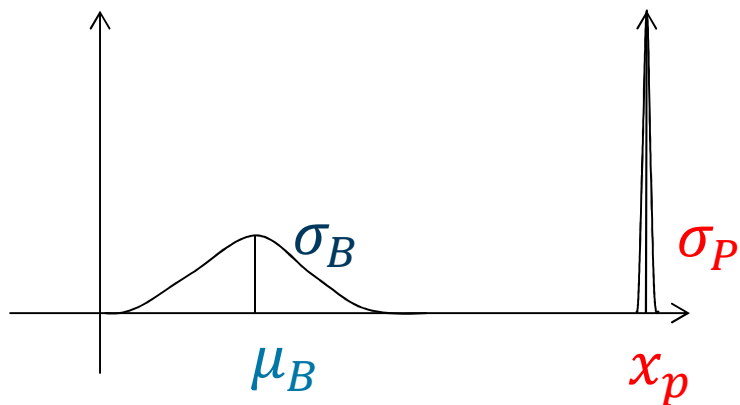
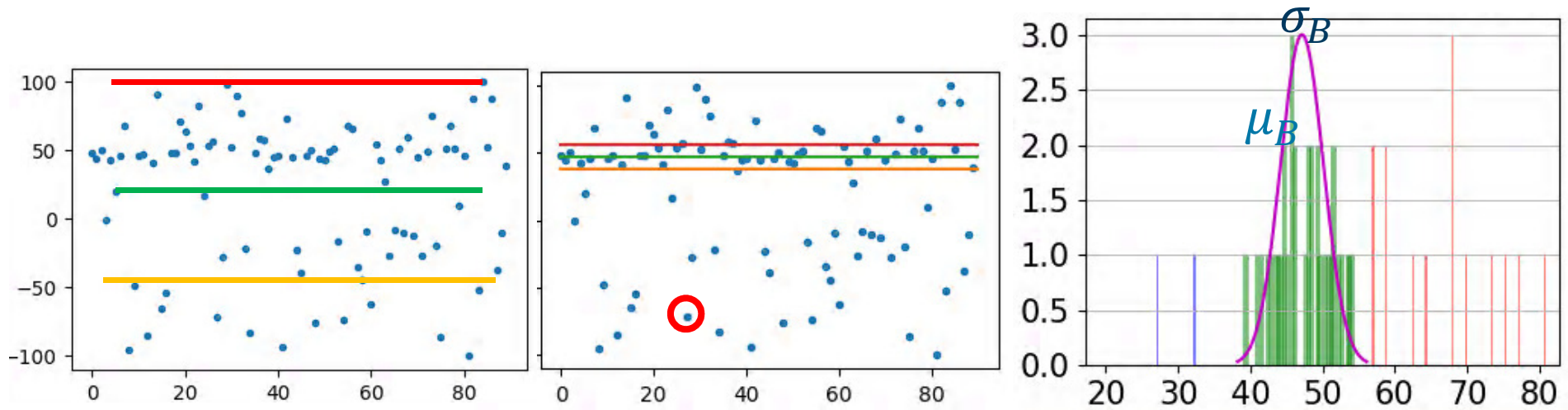
# Abnormal behavior



# Measures of pixels behavior

- Analog and Digital offsets in the dark
- Analog and digital variations in the dark
- Showing poissonian probability density behavior
- All of these measures will be reduced to normal and abnormal statistics
- That is outlier detection is used during finding normal ones
- Those that are separable from the normal are bad.

# A Robust Gaussian Fitting library



$$\text{Separability or } SNR = \frac{x_p - \mu_B}{\sigma_p + \sigma_B}$$

How data is made:

```
RNN0 = 50 + 5*np.random.randn(30)
RNN1 = 200*(np.random.rand(60)-0.5)
testData = np.concatenate((RNN0, RNN1))
np.random.shuffle(testData)
```

# How to use the current implementation

- The current implementation is for research and development
- It is basically a C file
  - And a Python3 “ctypes” wrapper
  - The entire library has three main parameters:
    - What is the secure highest portion of data inlier? Example: 50% is very safe
    - The normalized cutoff threshold for a Gaussian, e.g.:  $3\sigma$  away from the mean
- Given a data vector with less than half outliers and a Gaussian

```
import RobustGaussianFittingLibrary as RGFLib

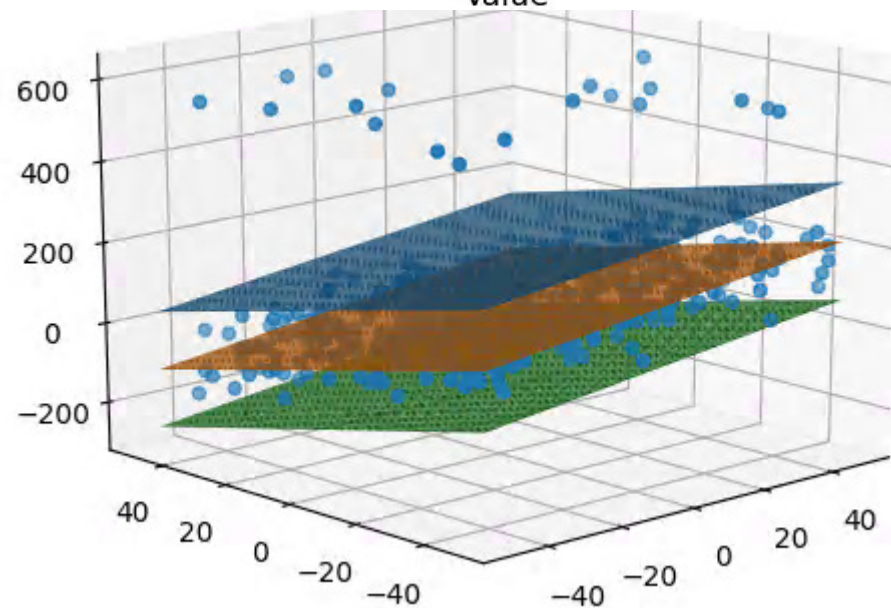
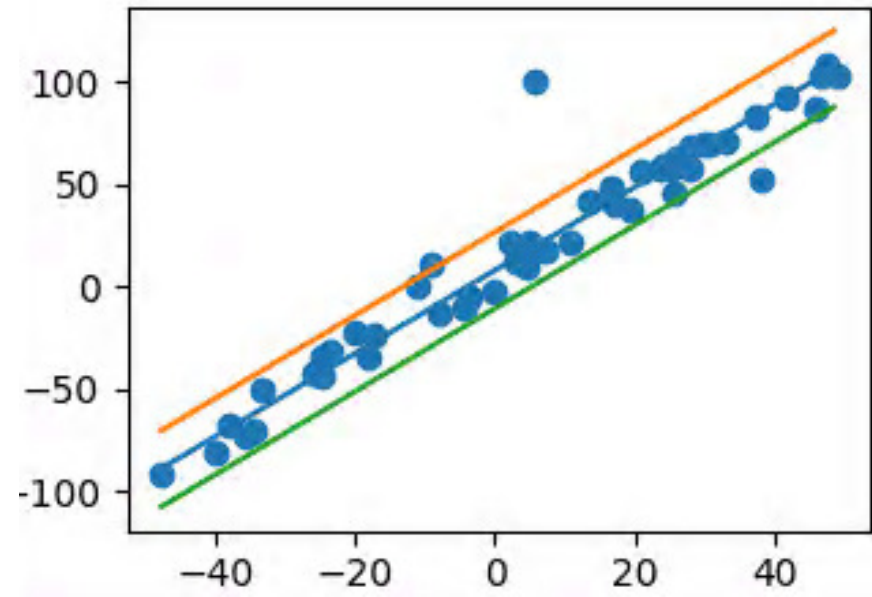
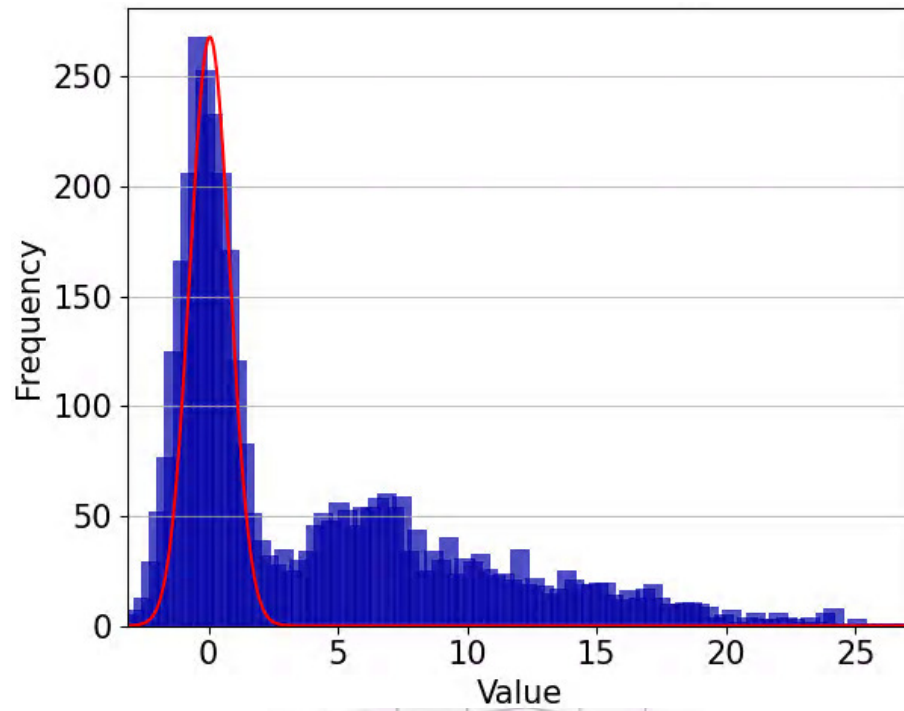
#Y = Mean + RSTD
RMean, RSTD = RGFLib.fitValue(data)

#Y = aX + RMean + RSTD
Ra, RMean, RSTD = RGFLib.fitLine(dataX, dataY)

#Z = aX + bY + RMean + RSTD
Ra, Rb, RMean, RSTD = RGFLib.fitPlane(dataX, dataY, dataZ)
```

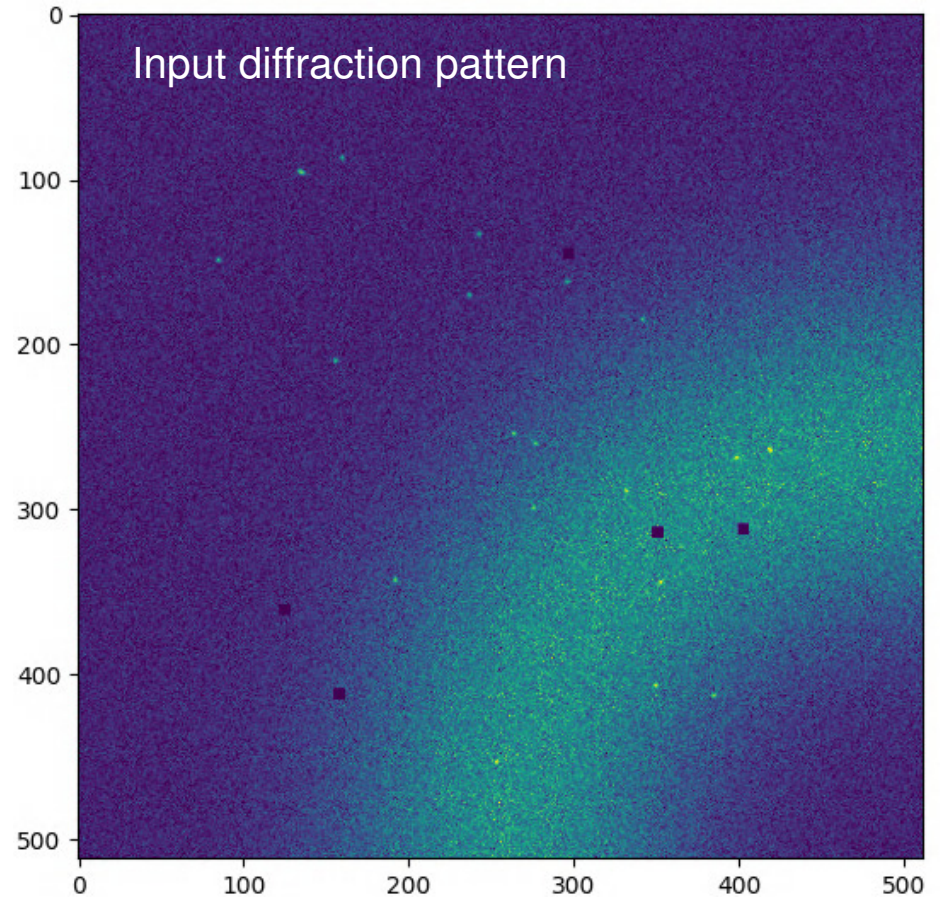
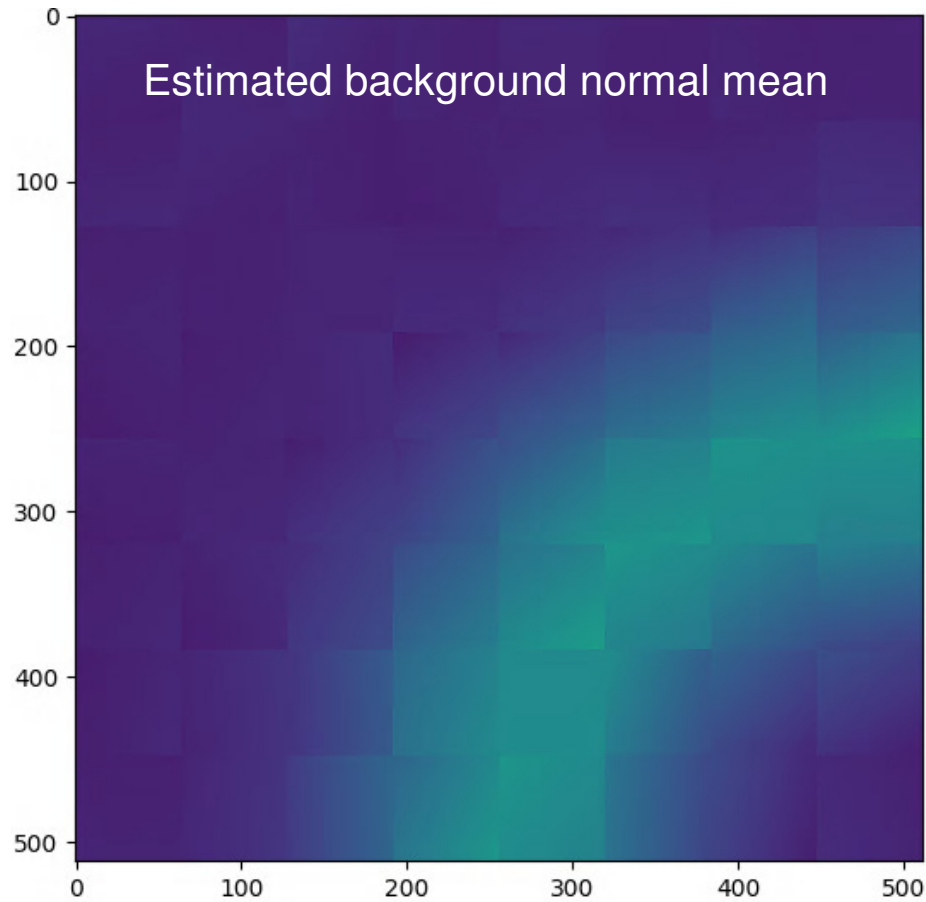
- Default parameters are the examples above

# More applications

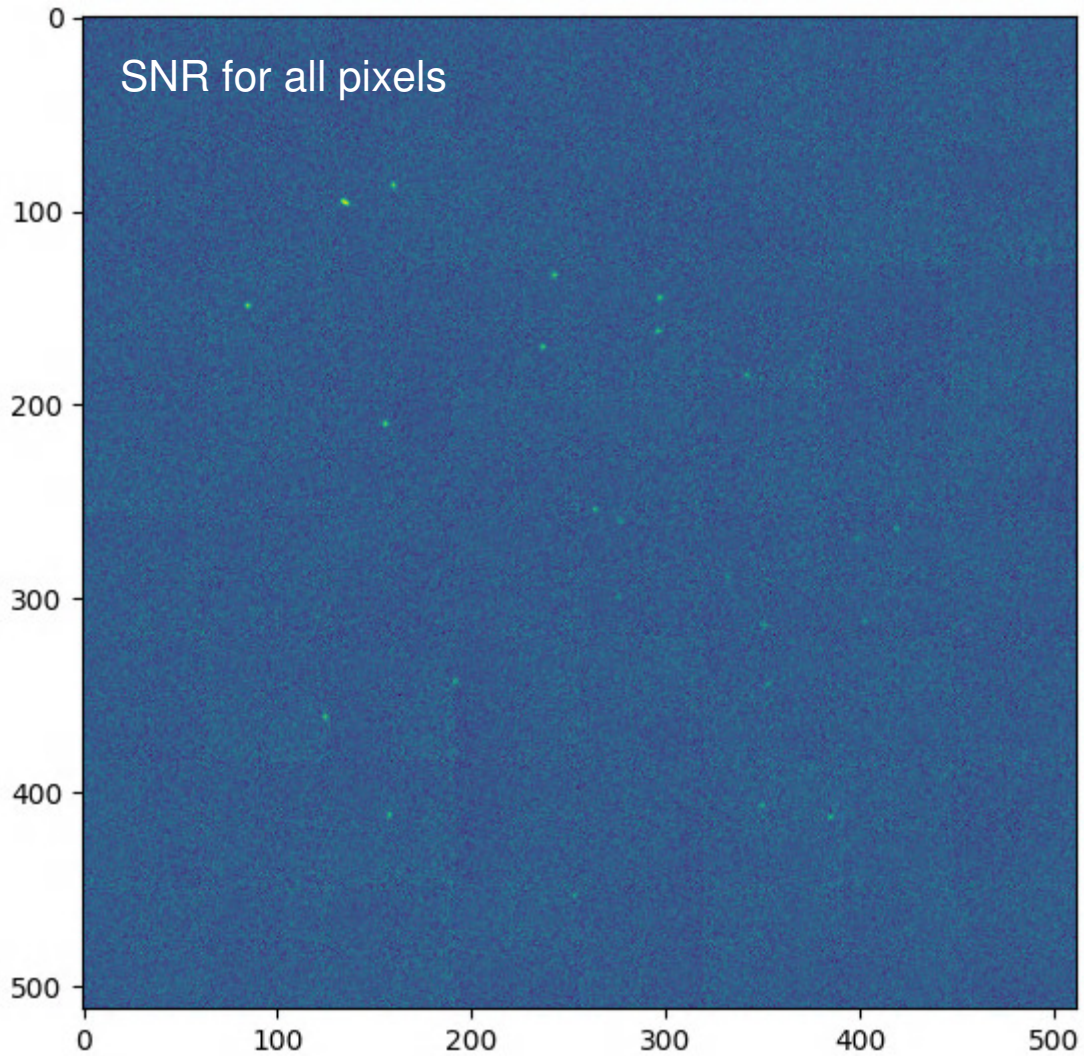




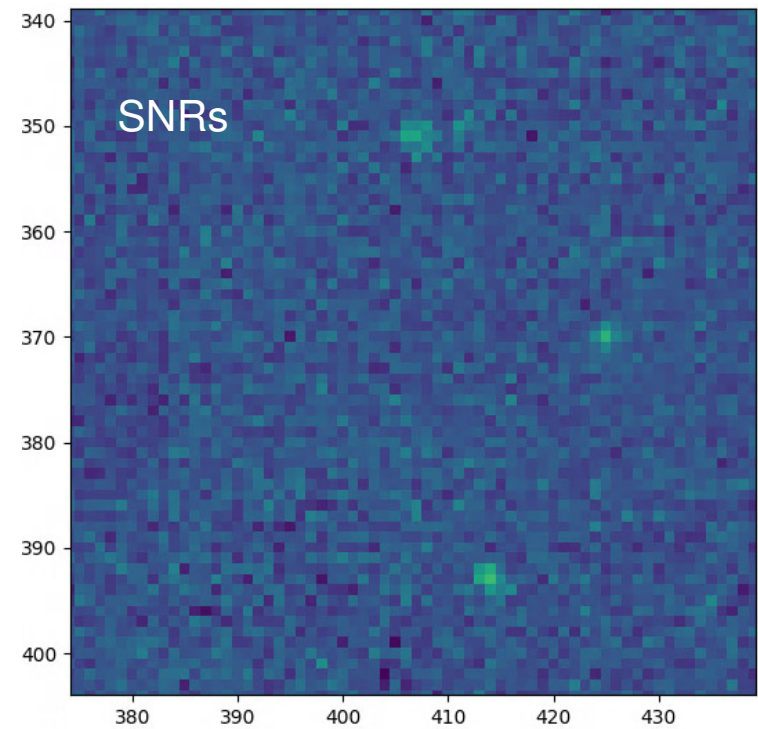
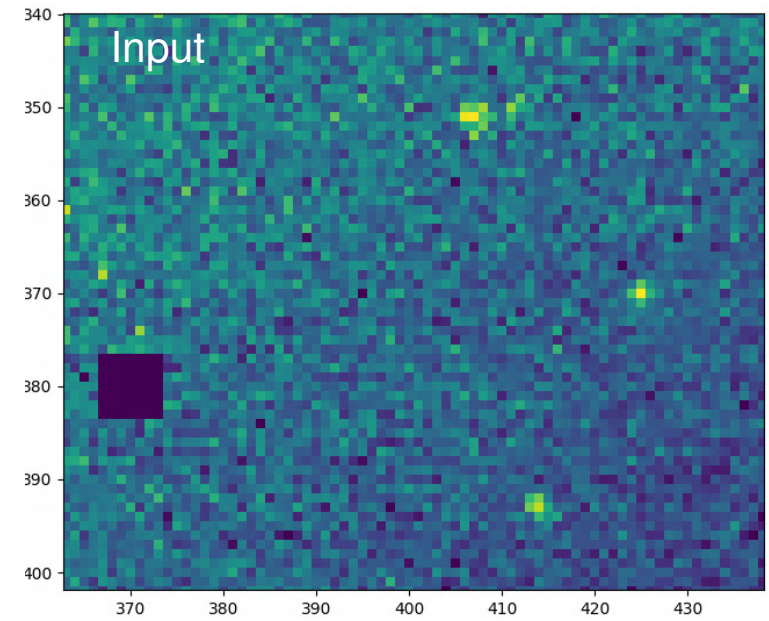
# Background estimation using RobustGausFitLib



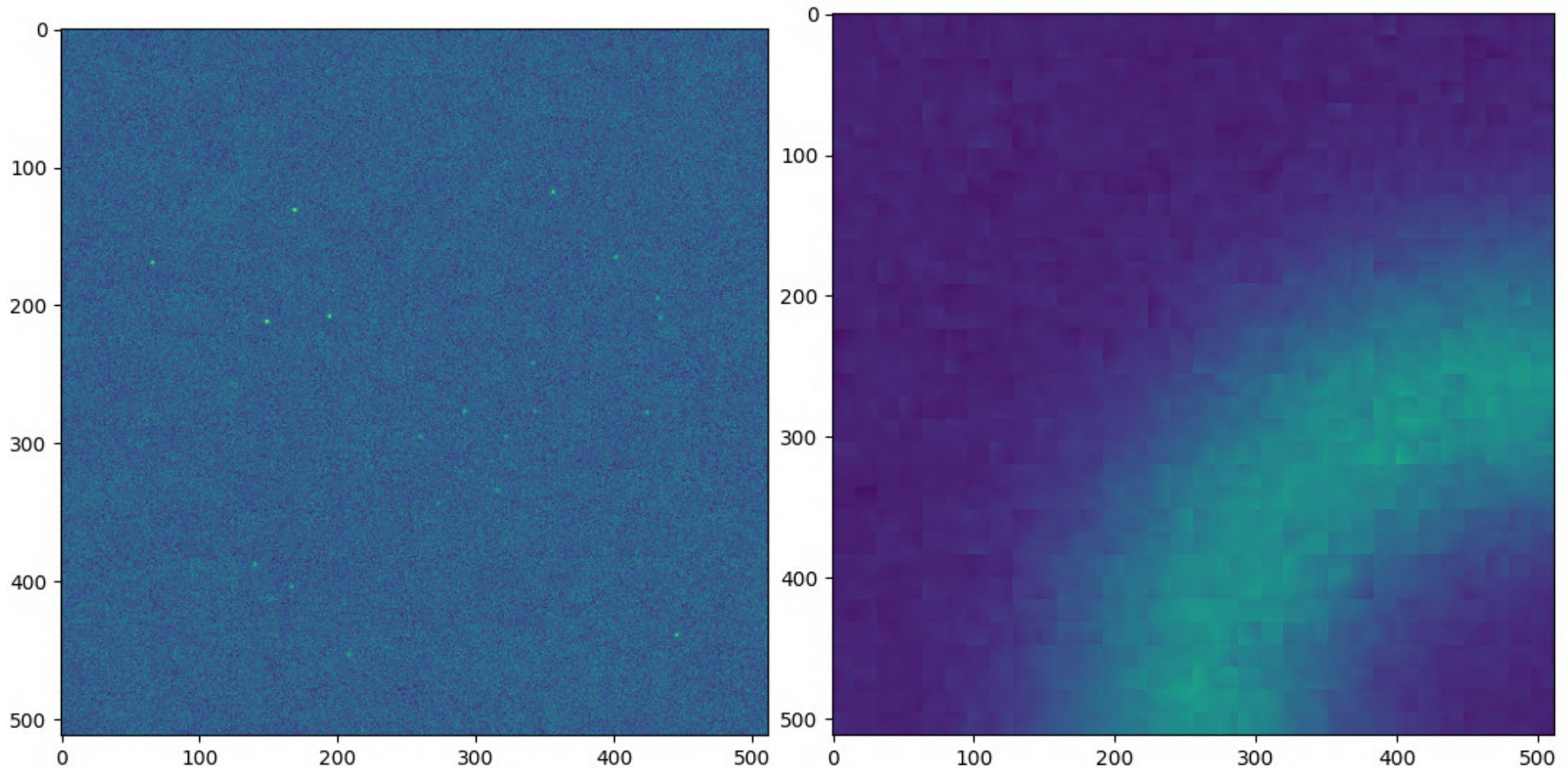
# Background subtraction



DESY.

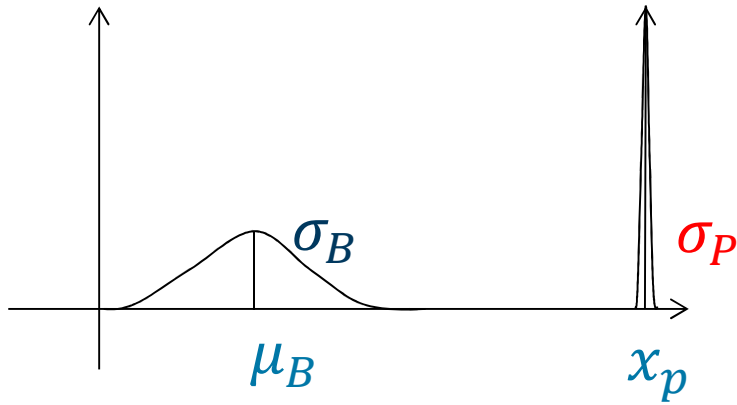


# More accurate background estimation By shifting and averaging multiple estimates



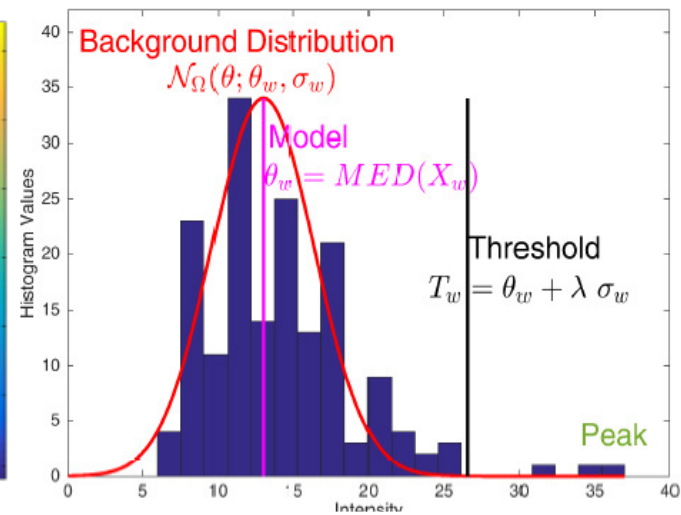
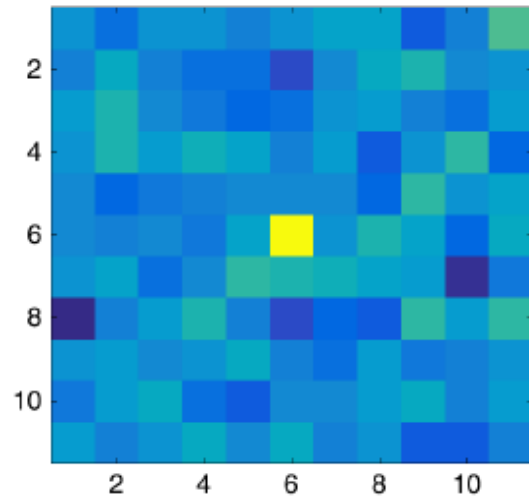
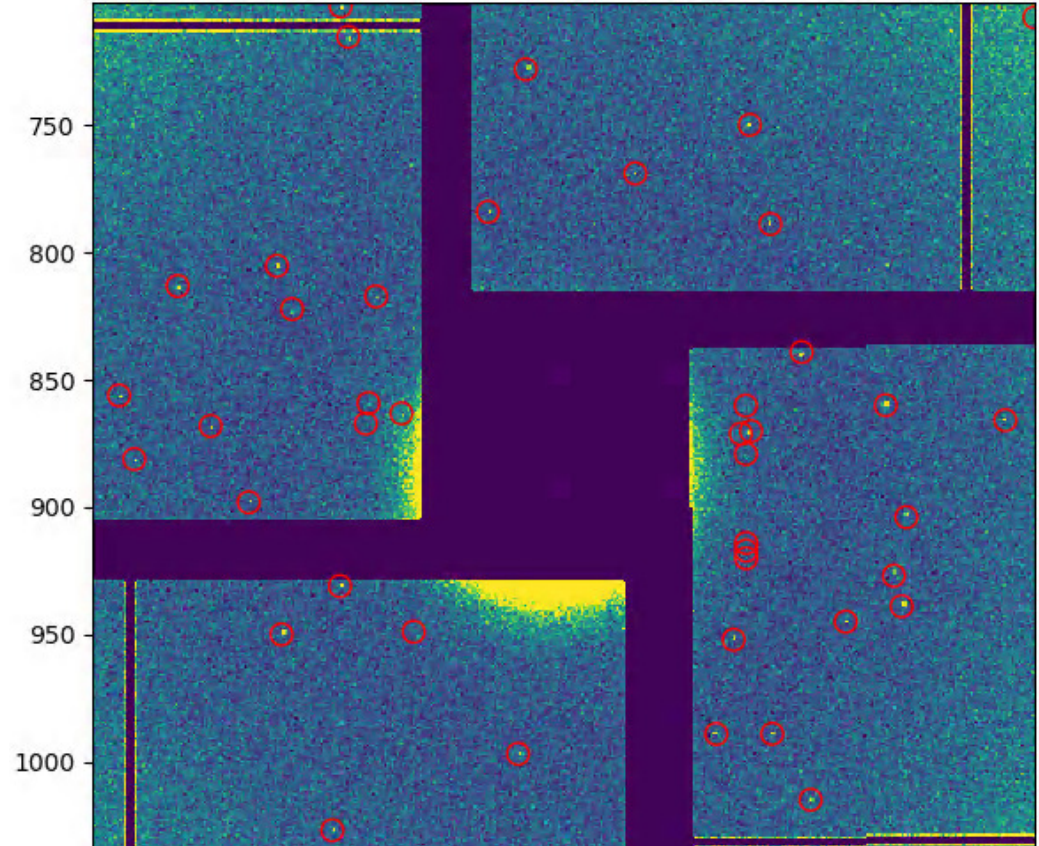
# Robust Peak Finder

<https://github.com/MarjanHJ/RobustPeakFinder>



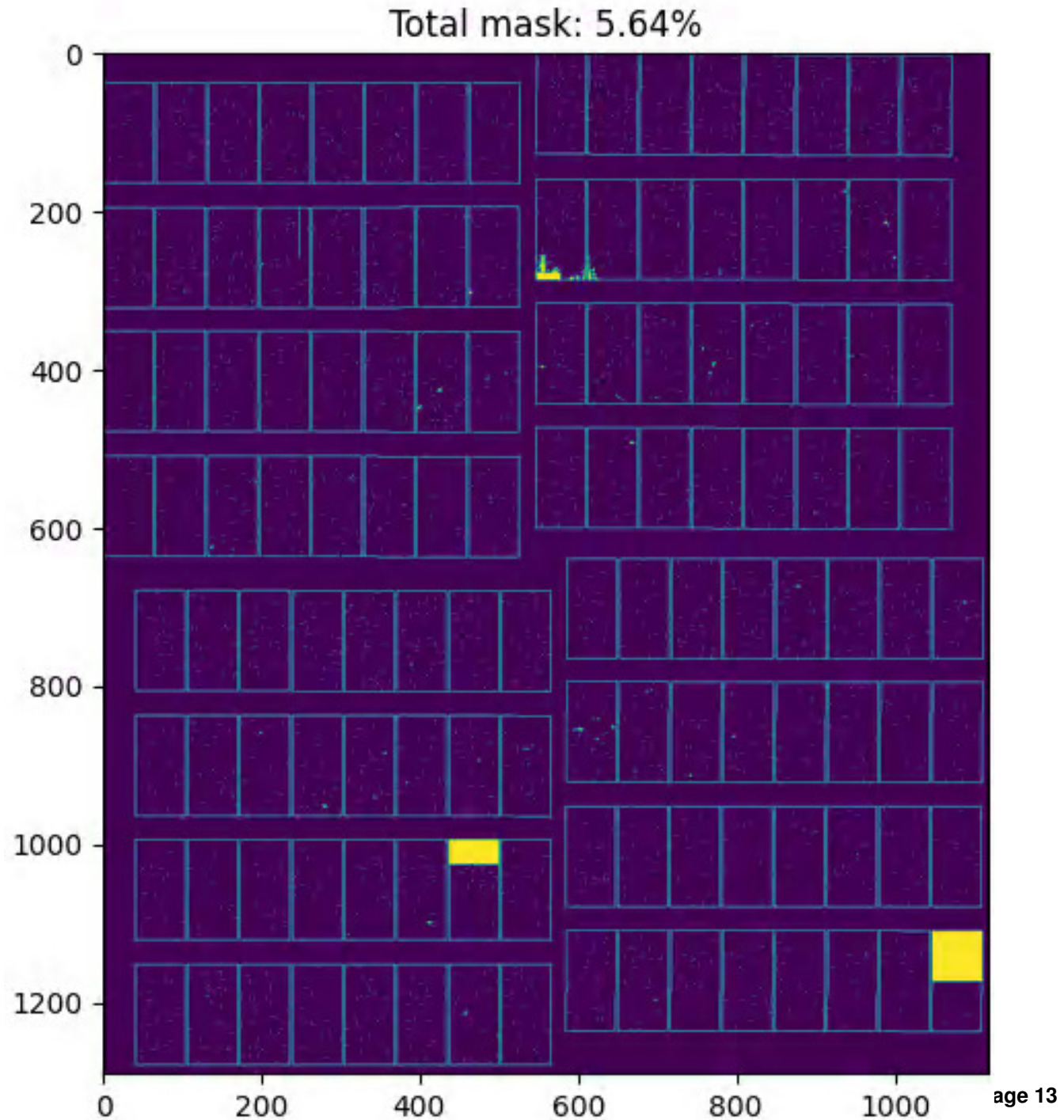
$$SNR = \frac{\mu_P - \mu_B}{\sigma_P + \sigma_B}$$

Run: 7, frame:13, SNR = 2.35, #Peaks=72



Overall  
(4.63 + 1%  
with T=6)

Memory  
cell 140



# Numerical results

- An experiment with Lysozyme
- In EuXFEL SPB with AGIPD 1M detector
- 5551744 images (3500 fps)
- Results for the effect of the mask on hit rate, indexing rate and partialator results (SNR, Rsplit and CC\* vs shells resolutions) on the above dataset for Robust peak finder and peakfinder8 in CrystFEL.
- During the tests, the only thing that changes is the mask
  - border of ASICs in AGIPD are always masked.
- For both methods
  - Acceptable SNR > 6.0
  - Hit threshold: 20 Bragg peaks
- By tapping into EuXFEL proc files, a mask can be found, used here for a non-conclusive friendly comparison

# Rates

## Out of 5551744 images

- Robust peak finder:
  - border mask:
    - Hits: 5170409
    - Hit rate: 93%
    - Indexed images: 1109294
    - Indexing rate: 21.45%



- Proposed masked
  - Hits: 1374927
  - Hit rate: 25%
  - Indexed images: 884329
  - Indexing rate: 64.31%

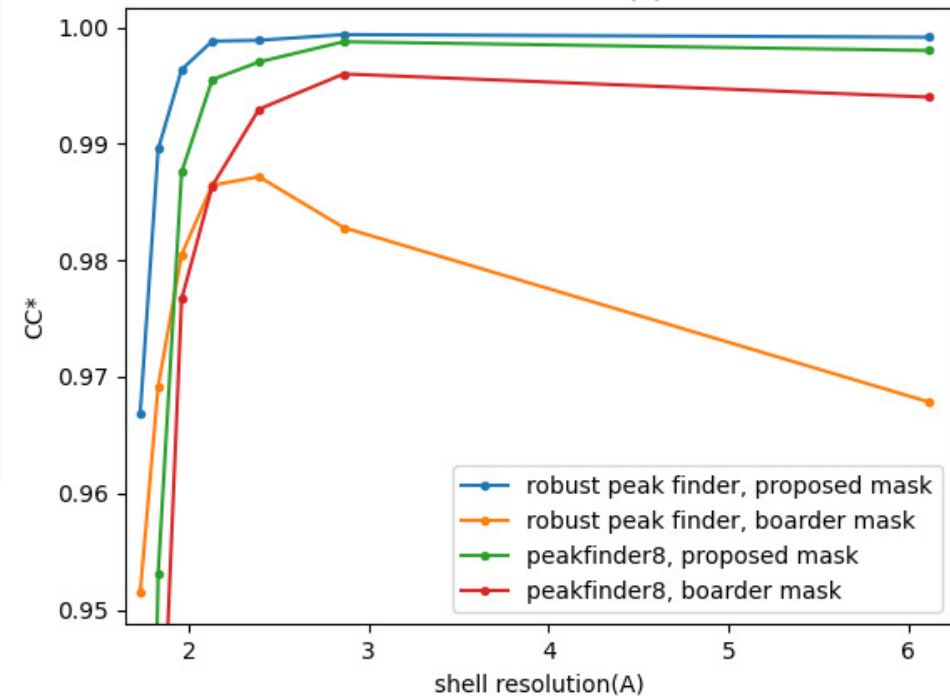
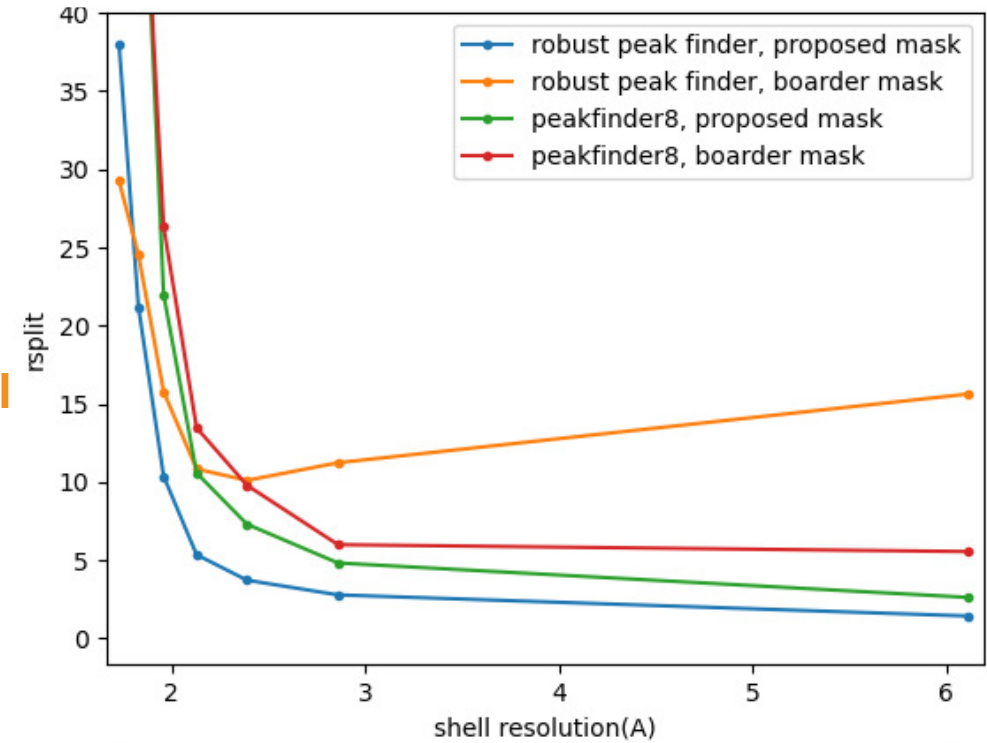
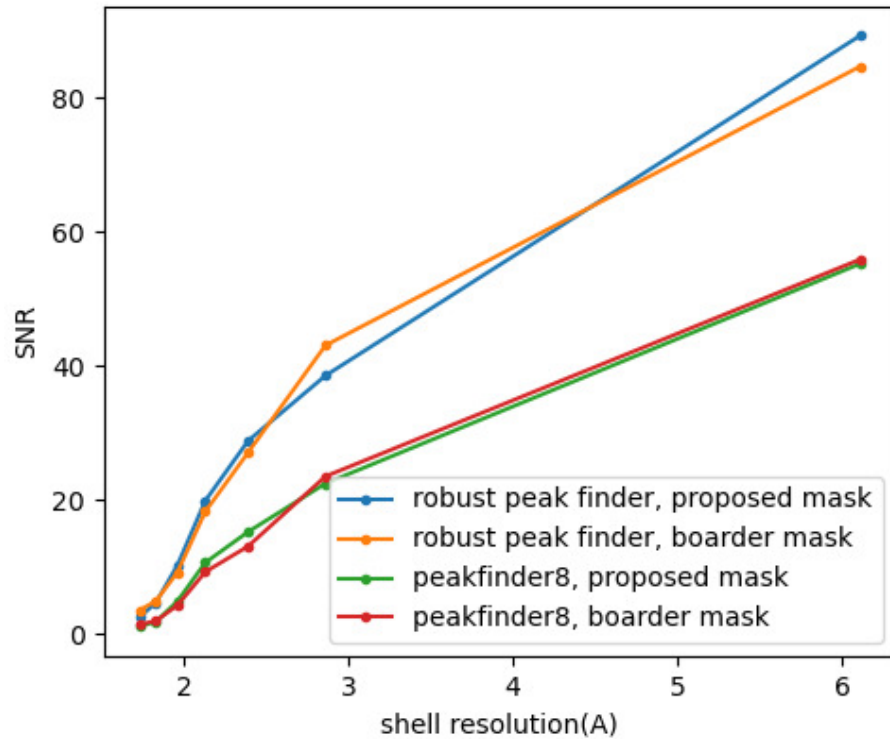
- CrystFEL peakfinder8
  - border mask:
    - Hits: 1432622
    - Hit rate: 26%
    - Indexed images: 546705
    - Indexing rate: 38.16%



- Proposed masked
  - Hits: 567850
  - Hit rate: 10%
  - Indexed images: 427433
  - Indexing rate: 75.27%

# Further in the crystallography pipeline

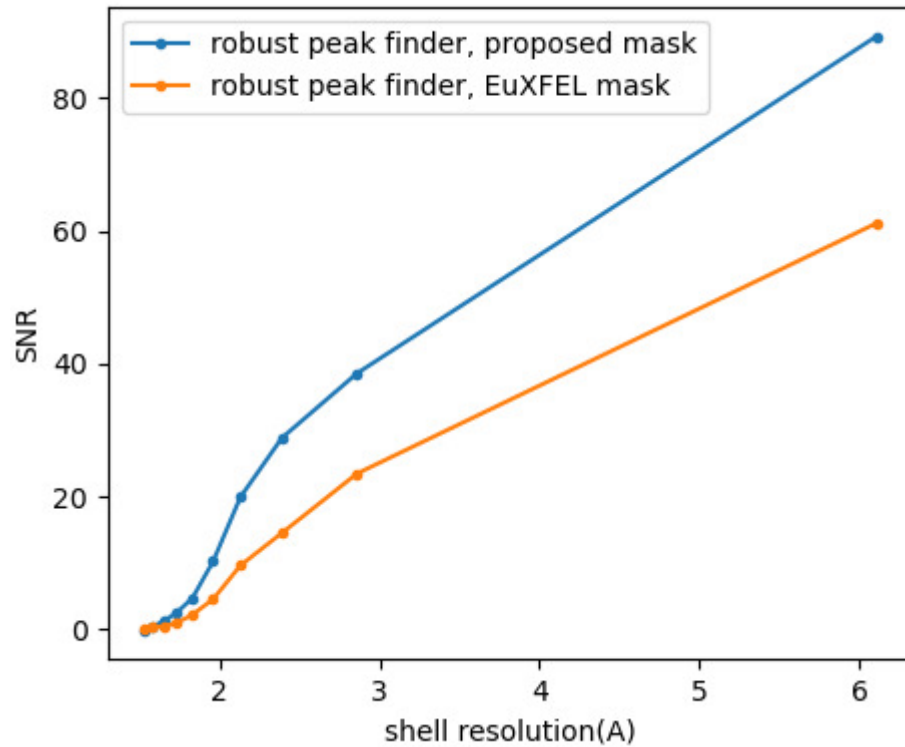
By running `partialator` and `check_hkl`



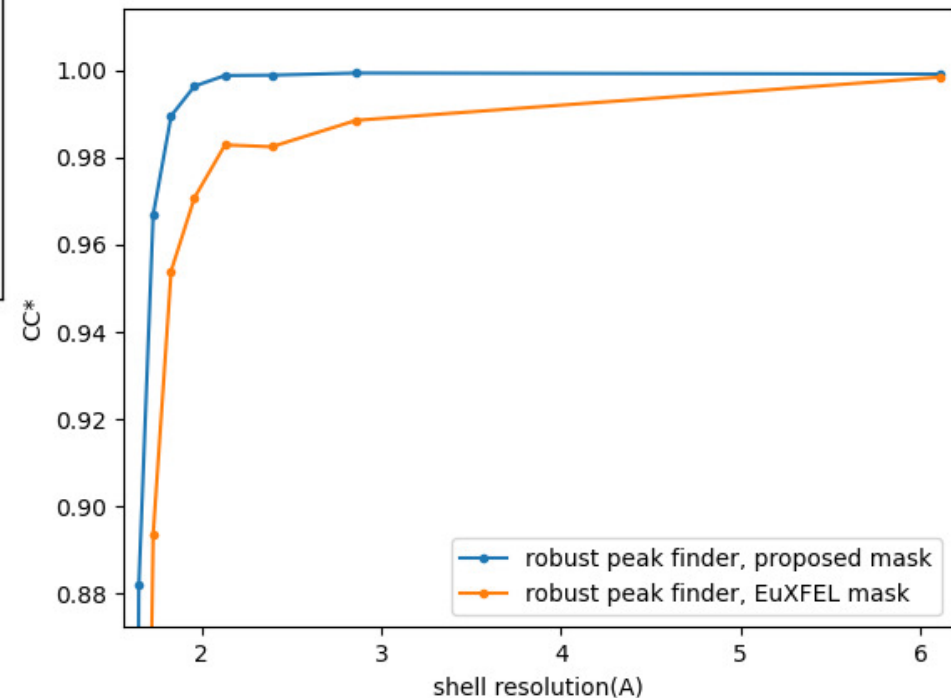
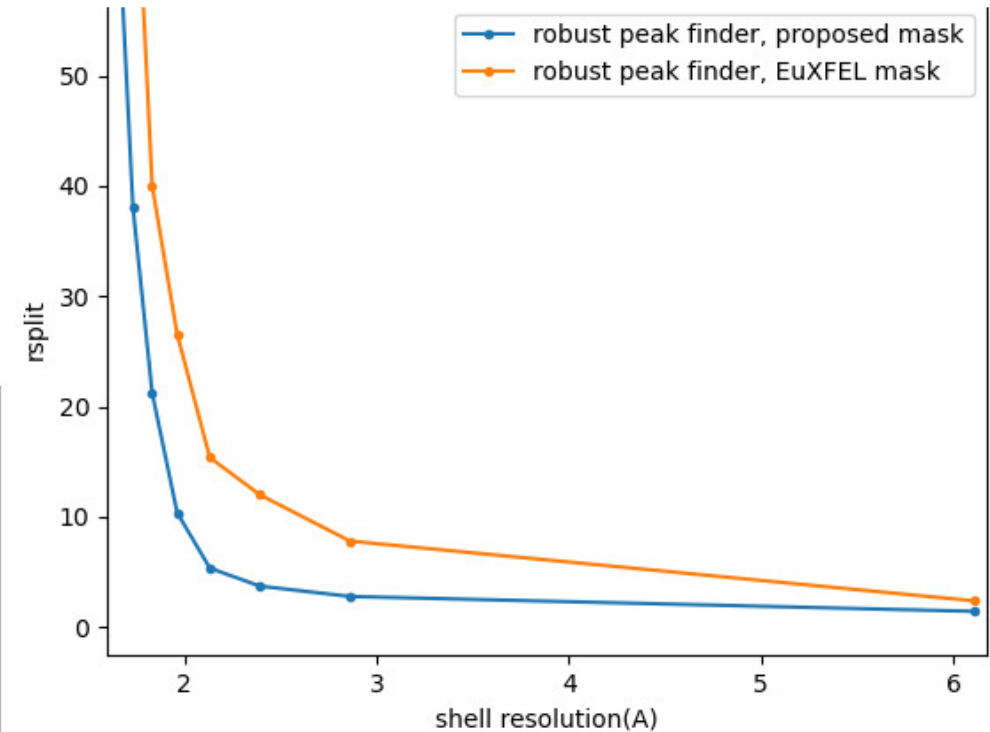


# A mask in EuXFEL proc files

A non-conclusive set of results



Hits: 2336085  
The hit rate: 42%  
Indexed are: 482433  
The indexing rate: 20.65%



# Thank you

## Contact

**DESY.** Deutsches  
Elektronen-Synchrotron

[www.desy.de](http://www.desy.de)

Alireza Sadri  
CFEL-DESY  
E-mail [Alireza.Sadr@desy.de](mailto:Alireza.Sadr@desy.de)