

Automatic pedestal tuning (After 2-bit offset calibration)

Varghese Babu

28th May, 2019.

- The idea is to tune the pedestal distribution of a module post offset-calibration so that it sits in a favourable position for zero suppressed data taking.
- We use a current source to shift the pedestal distribution to the left of the right of the dynamic range (0-255). Currently we use VnSubOut to shift pedestals to the right (higher values).
- We use a traditional measure-analysis-update procedure with 3 different scripts for each.

- The **measure script** performs pedestal scans for different values of the current source.
- The **analysis script** identifies the 'optimal values' of the current source from the pedestal scan.
- The **update script** then updates the best values to the system/database.
- The measurement and analysis are performed parallelly and both the measurement and analysis take ~1 minute each.

- **‘Optimal pedestals’?**

- We try to avoid ‘bad pedestal region’ and try to minimize the number of pixels in this region, also we try to have the distribution as low as possible so that we have enough room to describe the signal.

- We minimize a figure of merit

FOM = number of ‘bad pixels’ + (w * center of gravity of pedestal distribution in ADU)

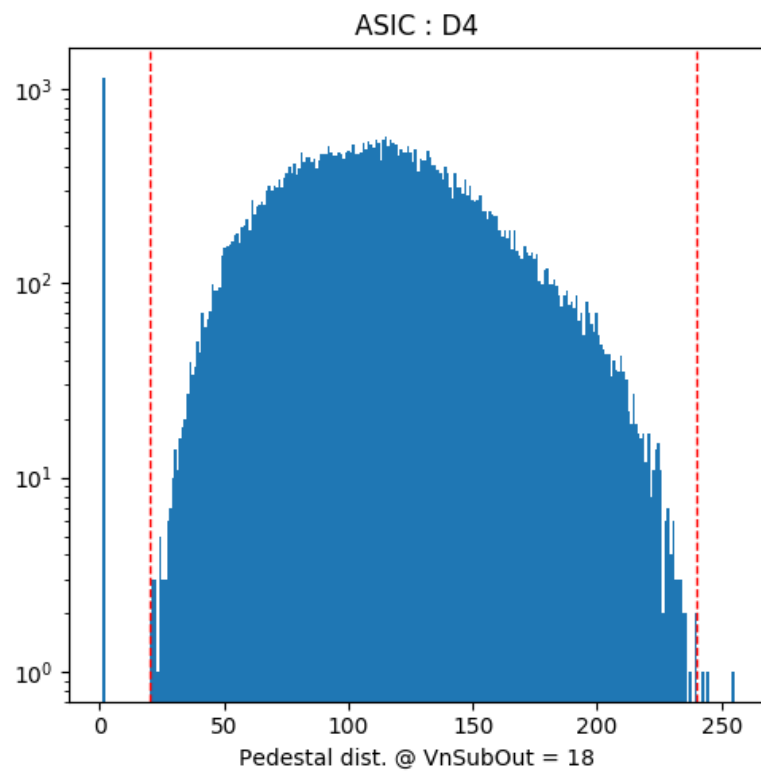
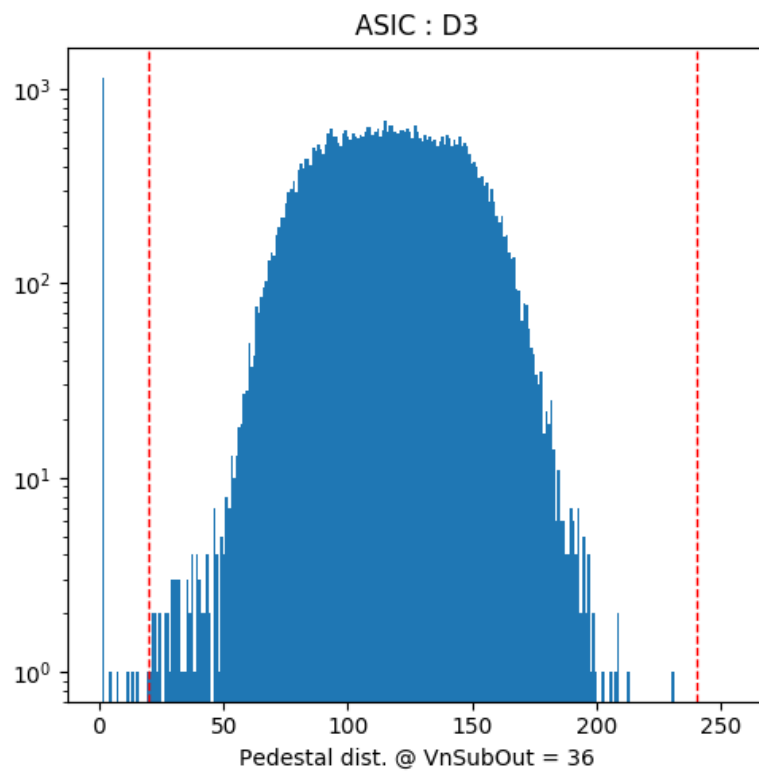
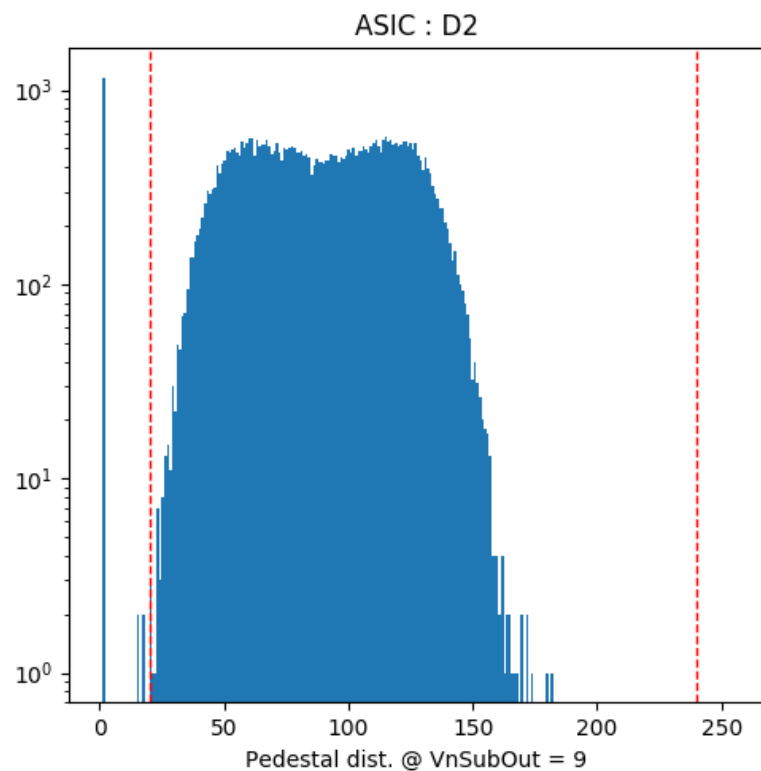
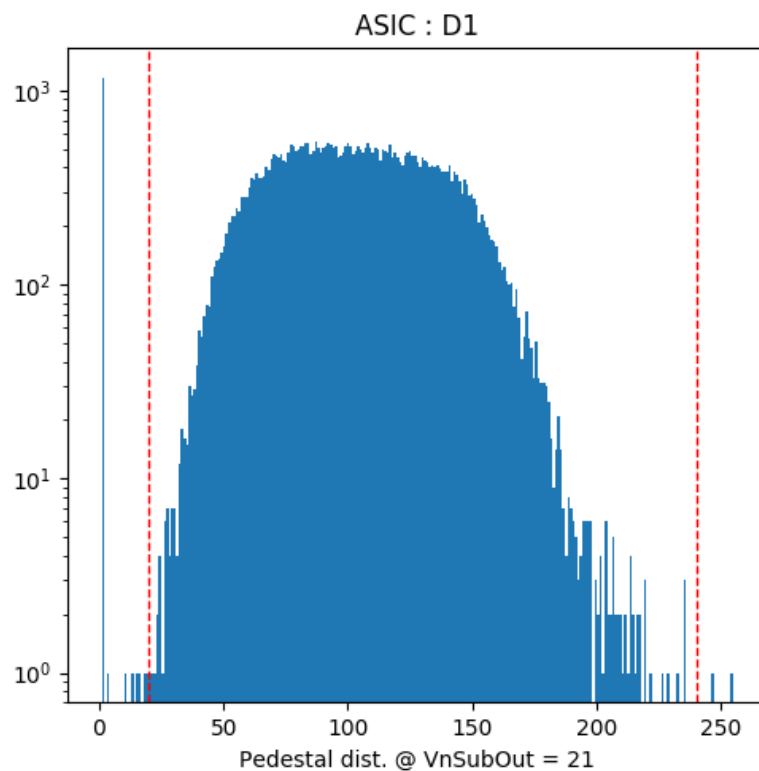
- w is a weight factor we try to get from experience.

- In the tests at DESY, w was set to 0.1, 1, and 10 and it looks like $w = 1$ seems to work quite well.

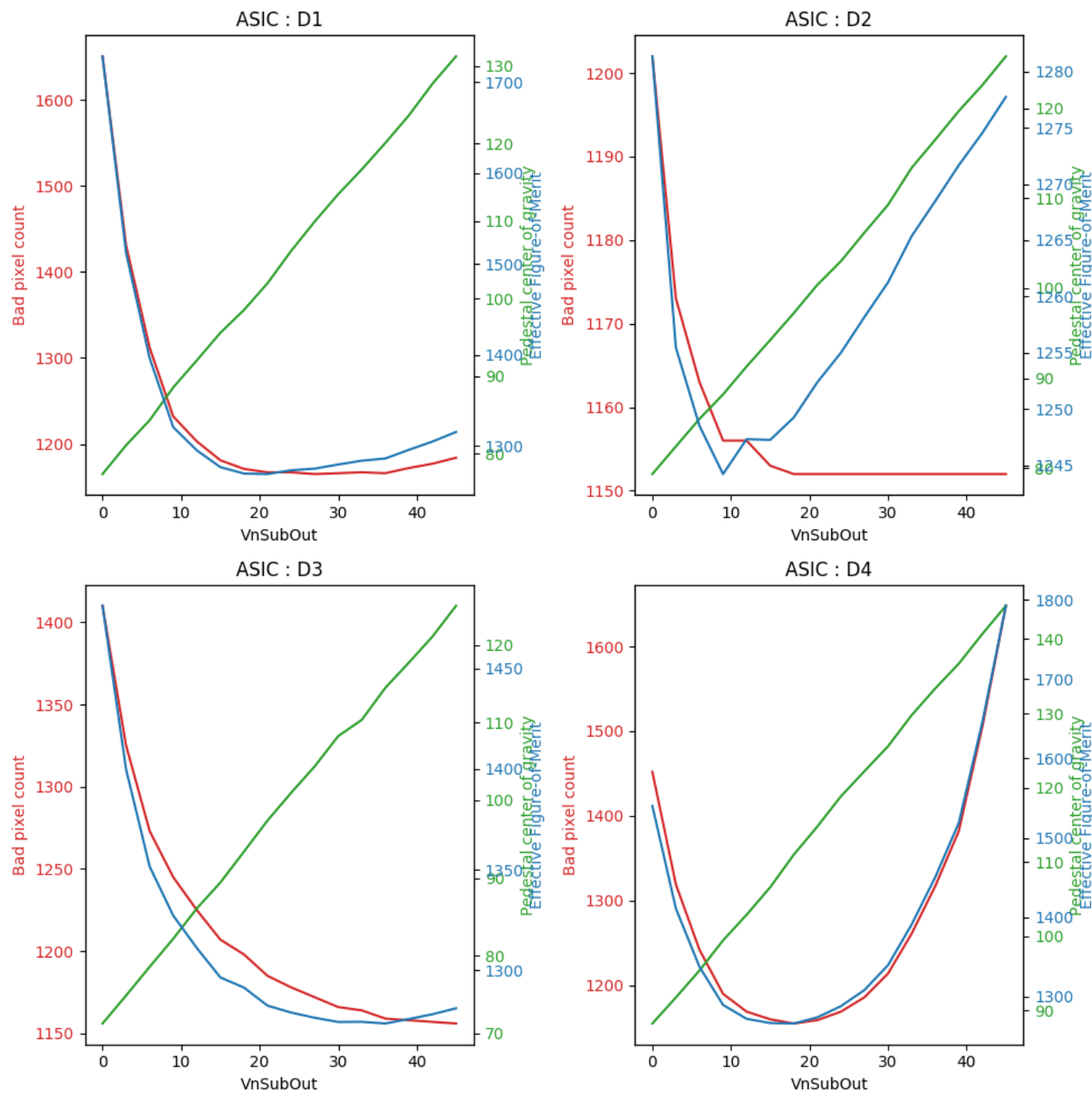
- ‘Bad pixels’ are ones having a pedestal value below 20 ADU and above 240 ADU (for plots in the next two slides)

DESY Module 1021

$w = 1$

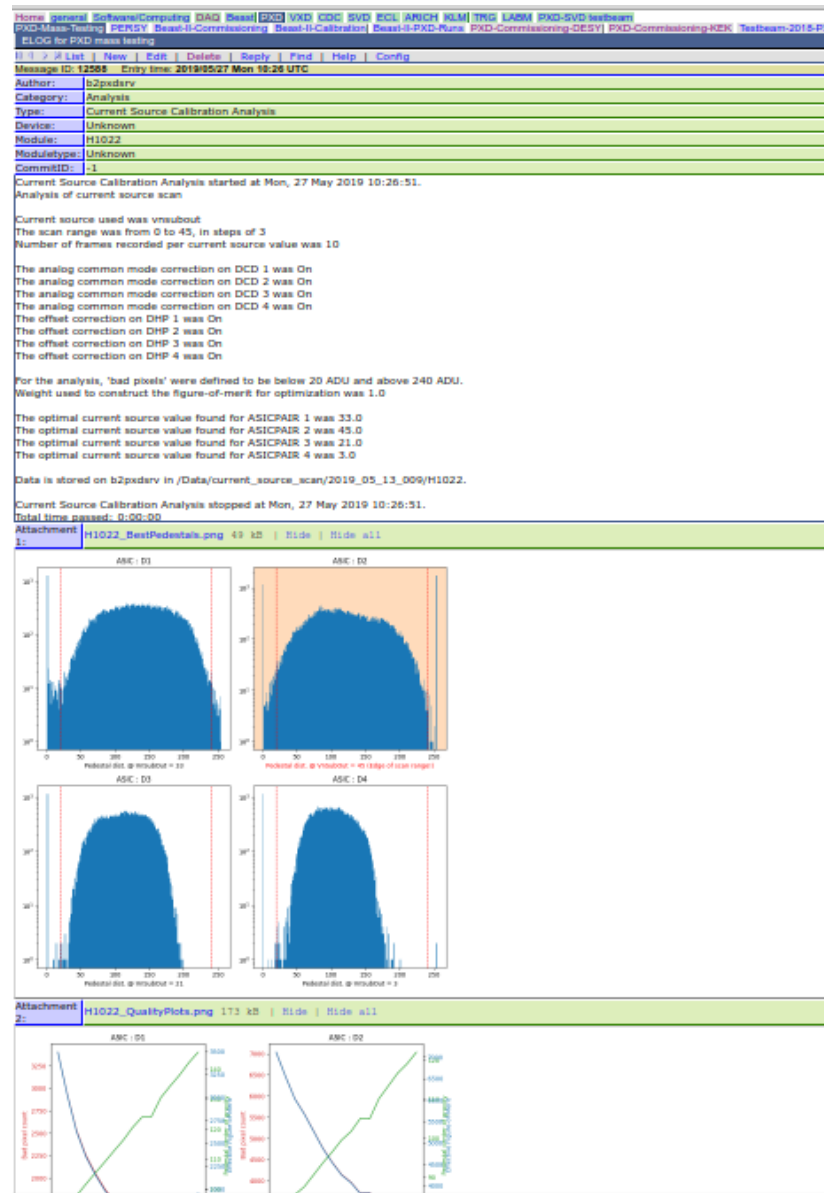


1021 quality plots



Elog entry

- There is an option now to create elog entries.
- For example, <https://elog.belle2.org/elog/PXD-Mass-Testing/12588>

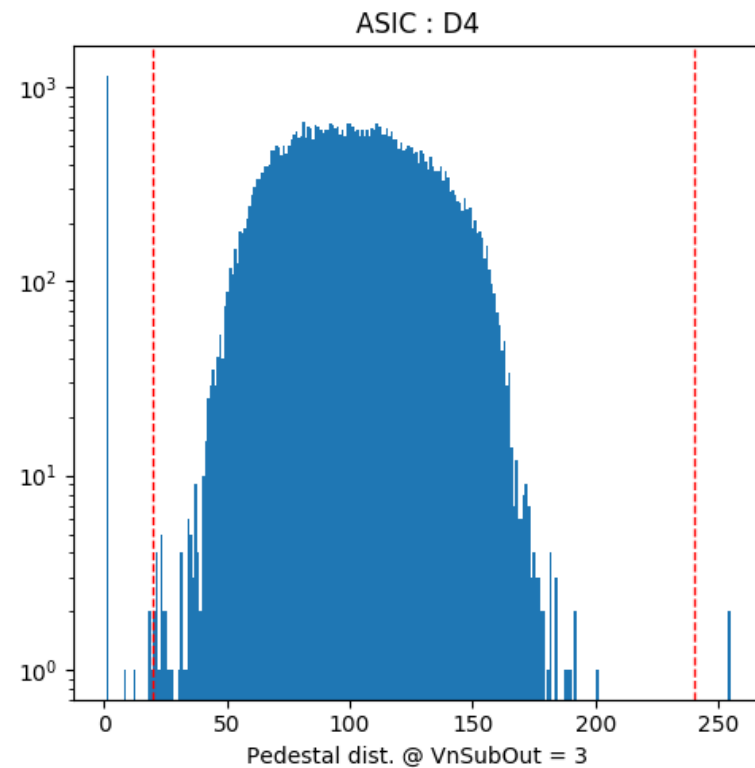
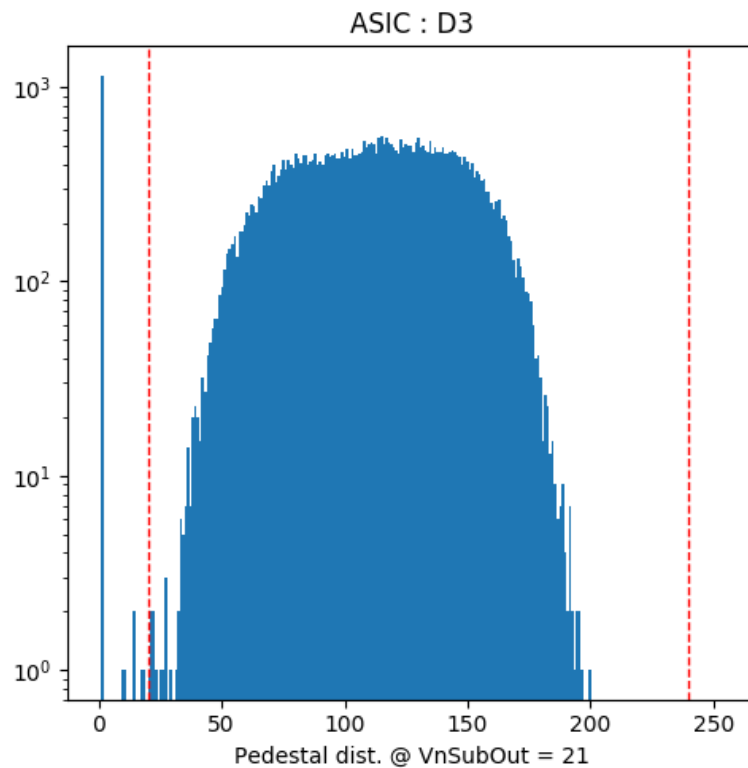
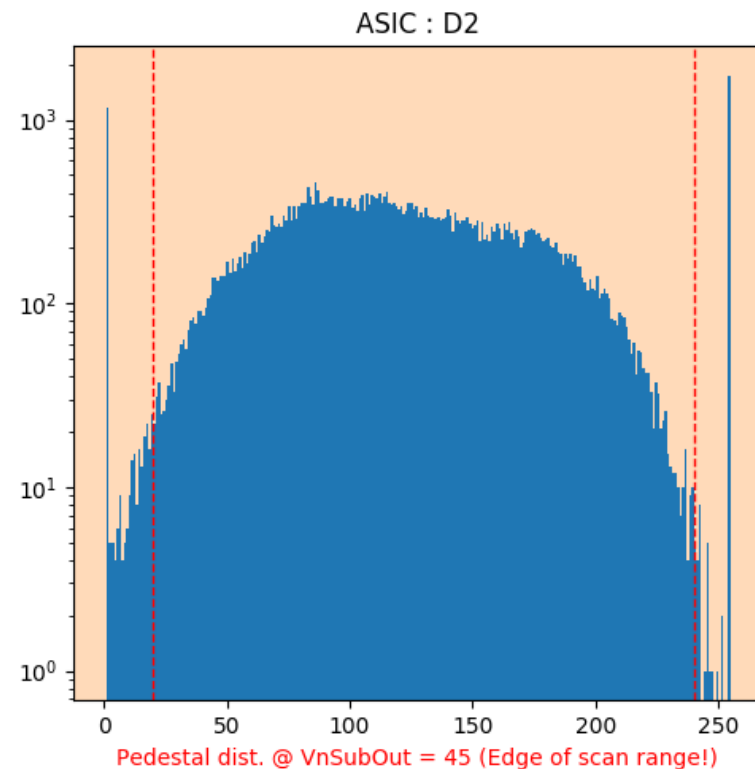
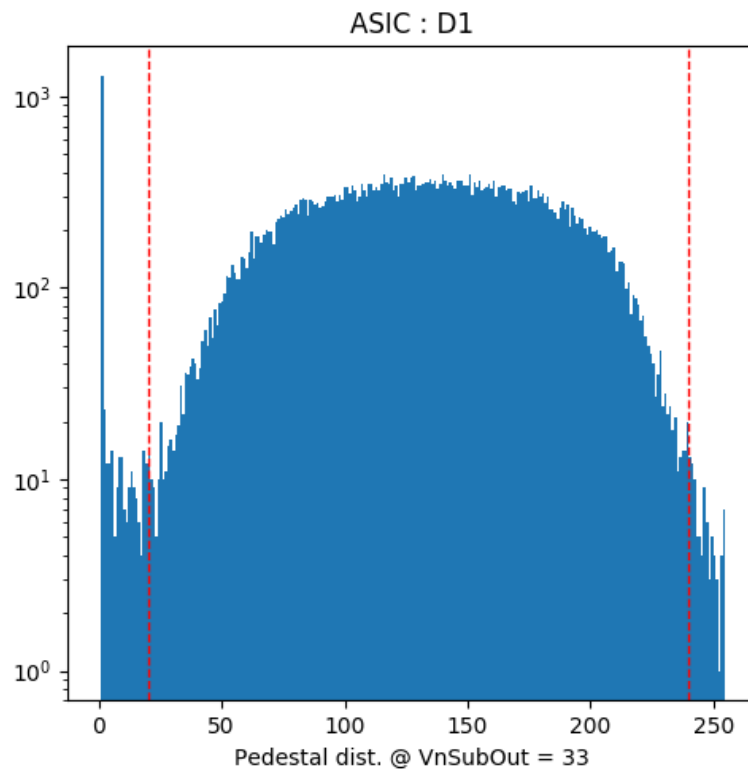


- The lower threshold for 'bad pixels' is set to 20 ADU. But maybe this lower limit can be determined for each module (as suggested by Florian)?
- For wide pedestal distributions, the upper-threshold would also play a part in the optimization
 - $255 - 7(\text{zs-threshold}) = 248$ is a hard limit as pedestals above it would anyway appear dead.
 - 240 is the value taken currently, so that even the ones at 240 can hold zs hits in the range 8-15 ADU (a bit arbitrary...).
- The code resides in the **feature/auto_pedestal_shifts** branch
- Hope to clean up a bit and create a pull request soon.

Backup

DES Y Module
1022

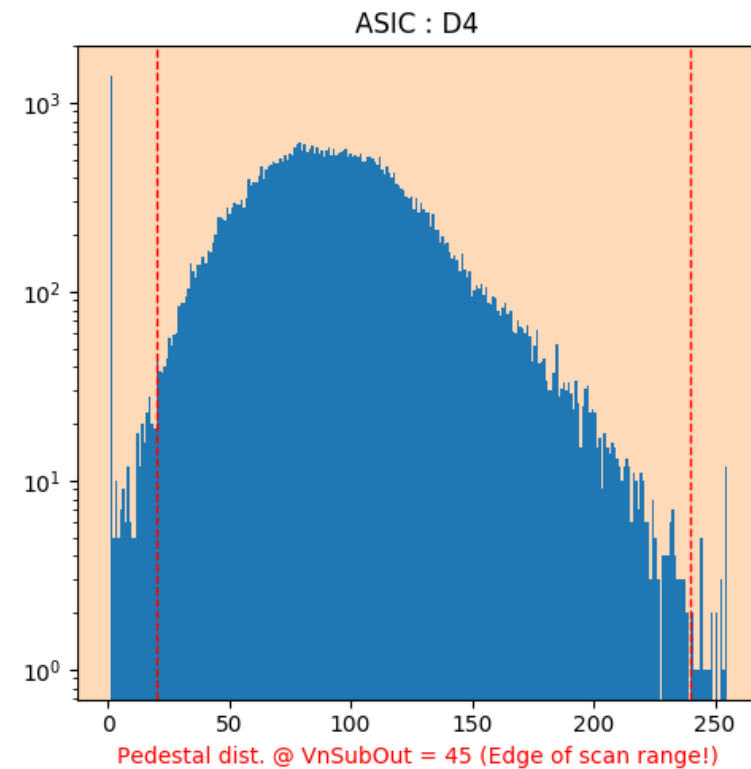
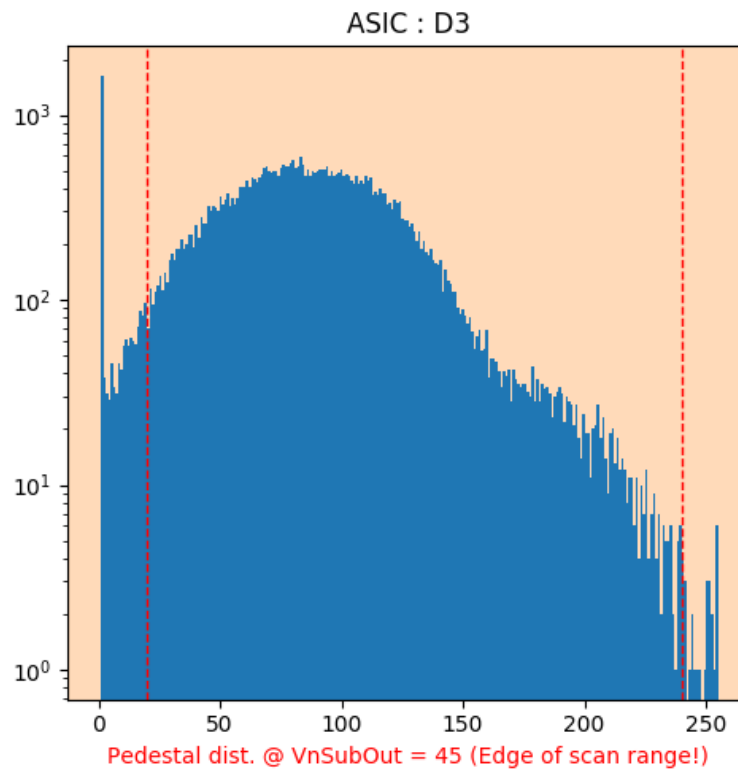
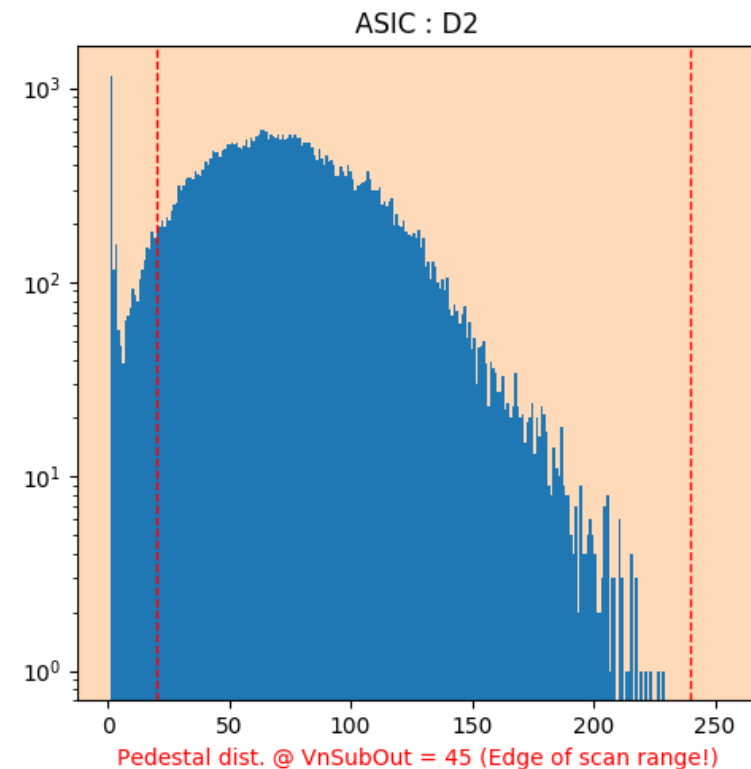
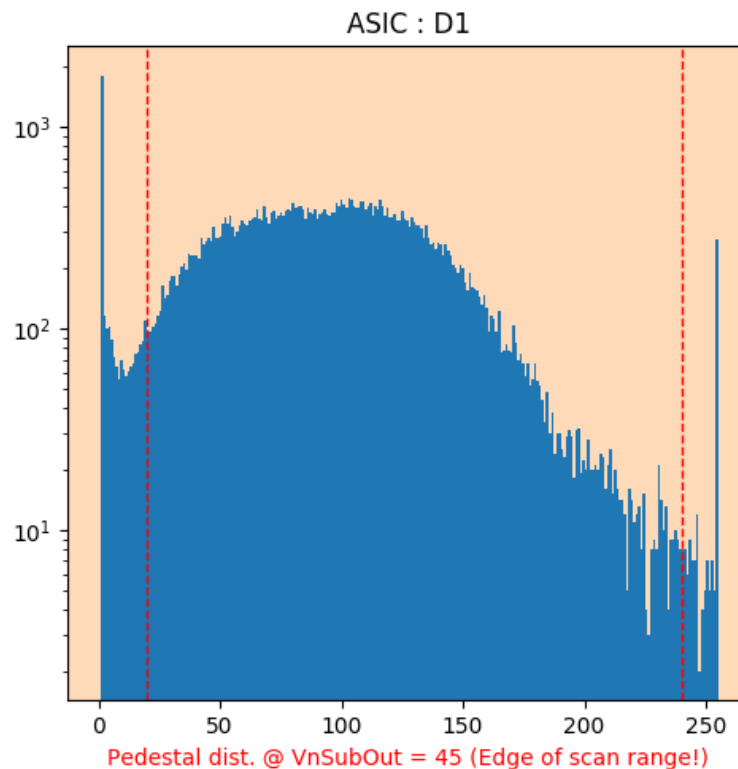
$w = 1$



Backup

DESY Module
1032

$w = 1$



Backup

DES Y Module
1031

$w = 1$

