simulation tools

Philipe Mota

CBPF

June 5, 2020

goal

- it is essencial to have a good control of the noise and dark current of the images since they affect directly detection threshold
- a good understanding of the behavior of the dark current is paramount for an efficient event extraction focusing in reducing the fake events
- hardware approach using skipper CCDs is underway which promises to decrease significantly the readout noise allowing the proper identification of the dark current
- I have advanced in a parallel direction: developping tools to extract the maximum information from the existing images both 1x1 and 1x5

goal

- analytically, I have derived the following relations

$$m_1[\mathrm{DC} + \mathrm{Noise}] = g\lambda + \mu[\mathrm{Noise}]$$

$$m_2[\mathrm{DC} + \mathrm{Noise}] = g^2\lambda + \sigma[\mathrm{Noise}]^2$$

$$m_3[\mathrm{DC} + \mathrm{Noise}] = g^3\lambda$$

  where m_i are the i-th moments of the Poisson-Norm distribution corrected by its first moment, μ and σ are the parameters of the Norm noise, λ is the Poisson rate and g is the convolution shift which physically controls how many ADUs the distribution is shifted by the presence of one electron (dark current), g is the charge gain
- these relations are redundant and can be used to attest the quality of the estimations

goal

- in practice, we have to relate these quantities with sample estimations of the data
- The straightforward associations are proposed in a first glance

$$\mu[\mathrm{Noise}] = \mathrm{mean}[\mathrm{OS}]$$

$$\sigma[\mathrm{Noise}]^2 = \mathrm{var}[\mathrm{OS}]$$

$$m_1[\mathrm{DC} + \mathrm{Noise}] = \mathrm{mean}[\mathrm{AC}]$$

$$m_2[\mathrm{DC} + \mathrm{Noise}] = \mathrm{var}[\mathrm{AC}]$$

$$m_3[\mathrm{DC} + \mathrm{Noise}] = \mathrm{mean}\big(X - \mathrm{mean}[\mathrm{AC}]\big)^3$$

- however, the presence of modulations and other features of the image make weaken the quality of these estimations
- furthermore, the presence of the data requires a careful analysis since all these quantites will be shifted by the data

goal

I focus in 3 parallel ways to tackle the problem

- correcting the image
- using robust estimators
- energy-independent separation of data and dark current

tool for image analysis

I have developed an user-friendly – ask Carla ;-) – tool for this analysis in fact, this presentation is can be generated by the tool sample call

```
python Image.py analyse ImagePresentation/median "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --plot-sections --plot-spectrum
```

other functionalities

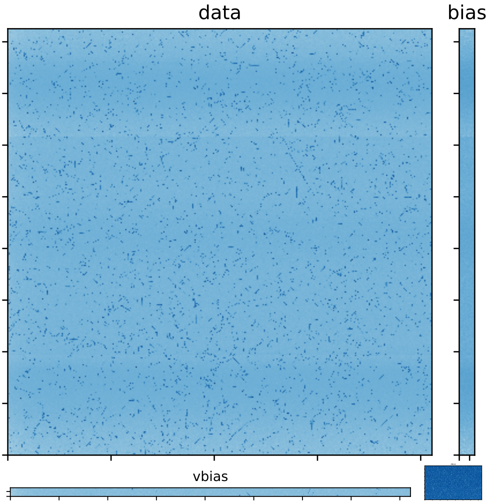
- read header
- simulate and get params
- extract hits (next week: comparison with offical extraction)

run locally

```
git init
git pull https://github.com/PhMota/CONNIEtools
python ImagePresentation.py
```

image imperfections

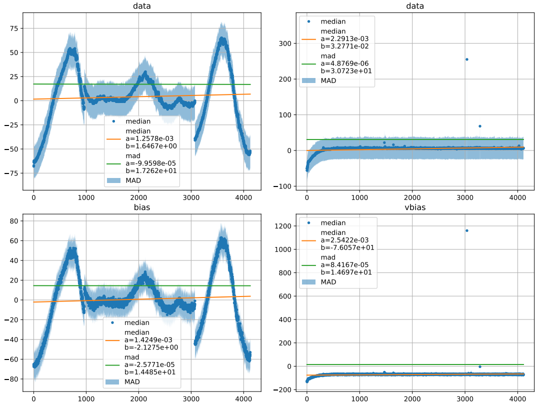
```
python Image.py analyse ImagePresentation/median
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --plot-sections --
    plot-spectrum
```



1x1 raw sections
vertical modulation is clearly visible

projections of the raw image

```
python Image.py analyse ImagePresentation/median
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --plot-sections --
    plot-spectrum
```



vertical modulation, horizontal modulation, hot columns

projections of the raw image

the MonitorViewer tool attempts to circumvent these imperfections, by estimating the quantities independently for each line and taking the mean over these results

$$\sigma = \mathrm{mean}\big(\mathrm{MAD}[\mathrm{OS}]_i\big)$$

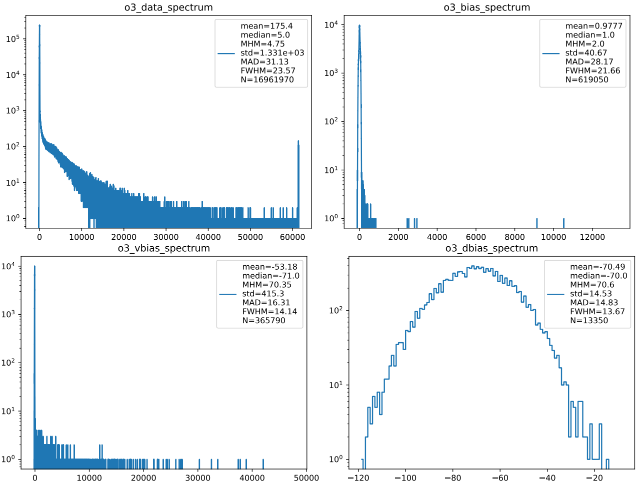
$$g\lambda = \mathrm{mean}\big(\mathrm{median}[\mathrm{AC}]_i - \mathrm{median}[\mathrm{OS}]_i\big)$$

$$g^2\lambda = \mathrm{mean}\big(\mathrm{MAD}[\mathrm{AC}]_i - \mathrm{MAD}[\mathrm{OS}]_i\big)$$

these quantities give consistent estimations for simulations generated with no(!) data

spectra of the raw image

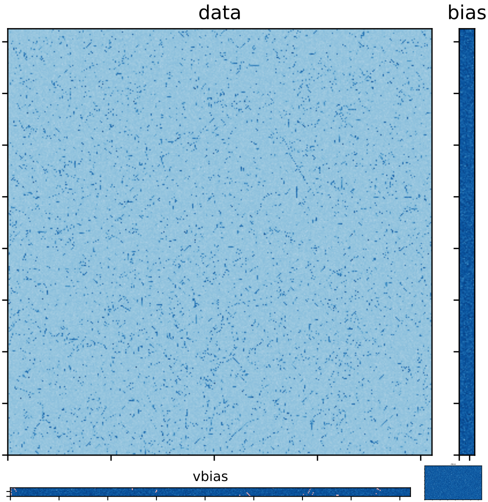
```
python Image.py analyse ImagePresentation/median
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --plot-sections --
    plot-spectrum
```



however, real distributions are crowded with outliers which heavily impair the capability of accurately estimating the parameters of the distribution

sides with line and col corrections

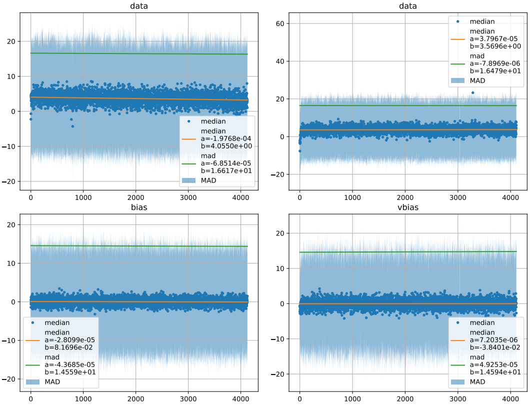
```
1    python Image.py analyse ImagePresentation/mean
       "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
       plot-sections --plot-spectrum
```



perform overscan subtraction by estimating the mean of the distribution line by line after removing the outliers then remove the vertical overscan modulation

projections with line and col corrections

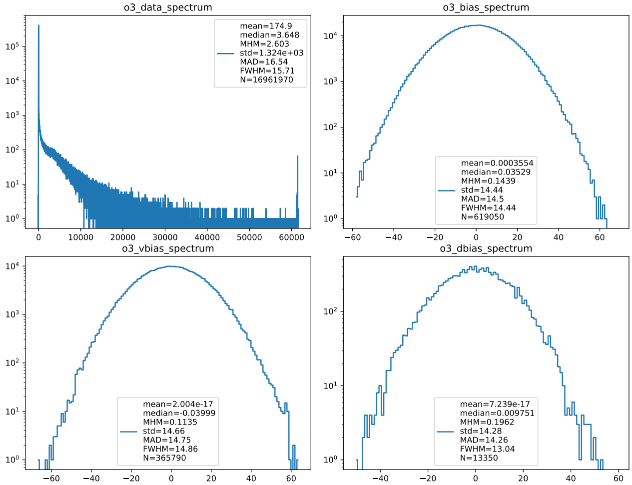
```
python Image.py analyse ImagePresentation/mean
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    plot-sections --plot-spectrum
```



data becomes quite stable on both lines and columns

spectra with line and col corrections

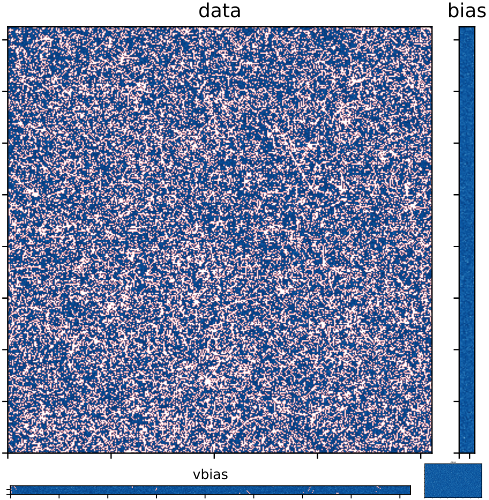
```
python Image.py analyse ImagePresentation/mean "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --plot-sections --plot-spectrum
```



this proceedure over the overscans succesfully removes their outliers allowing us to use redundant estimators to control the estimations

removed above 40ADU with border 3

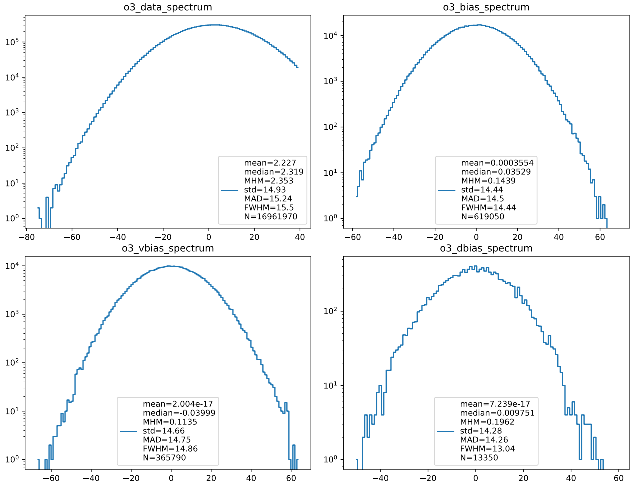
```
1    python Image.py analyse ImagePresentation/mean
         "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --remove-hits 40 3 --
         params-mode mean --plot-sections --plot-spectrum
```



in the data region we use a cluster removal algorithm and experiment with border of 3 pixels following the offical extractor approach

removed above 40ADU with border 3 spectra

```
1    python Image.py analyse ImagePresentation/mean
         "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --remove-hits 40 3 --
         params-mode mean --plot-sections --plot-spectrum
```



$$\sigma = 14.44$$

$$g\lambda = 2.32$$

$$g^2\lambda = 23.744$$

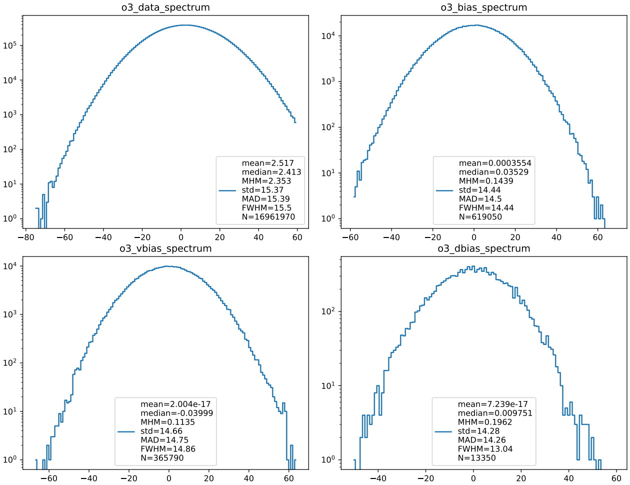
$$g = 10.23$$

$$\lambda = 0.23$$

perhaps 40ADU was too agressive since a large chunck of the Poisson-Norm distribution was also removed

$E < 60$ADU (+3 border) spectra

```
python Image.py analyse ImagePresentation/mean
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 60.0 3.0 --plot-spectrum
```



estimations

$$\sigma = 14.4$$

$$g\lambda = 2.41$$

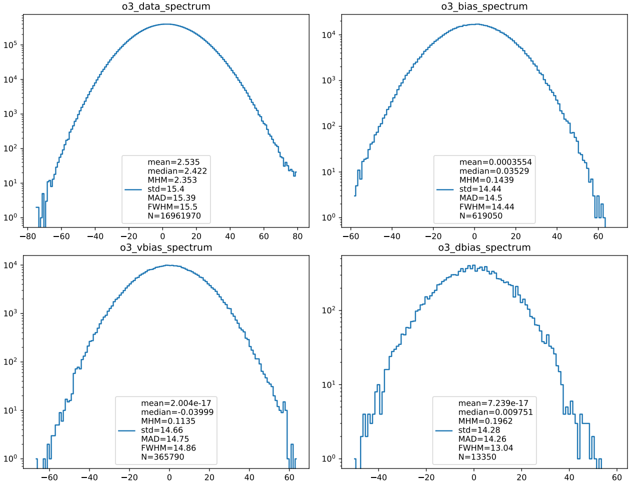
$$g^2\lambda = 26.6021$$

$$g = 11.04$$

$$\lambda = 0.22$$

$E < 80$ADU (+2 border) spectra

```
python Image.py analyse ImagePresentation/mean
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 80.0 2.0 --plot-spectrum
```



estimations

$$\sigma = 14.4$$

$$g\lambda = 2.42$$

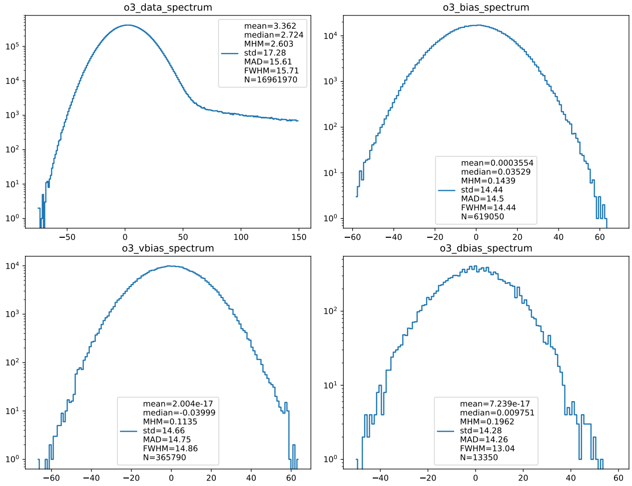
$$g^2\lambda = 26.6021$$

$$g = 10.99$$

$$\lambda = 0.22$$

$E < 150$ADU (+0 border) spectra

```
1 python Image.py analyse ImagePresentation/mean
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 150.0 0.0 --plot-spectrum
```



estimations

$$\sigma = 14.4$$

$$g\lambda = 2.72$$

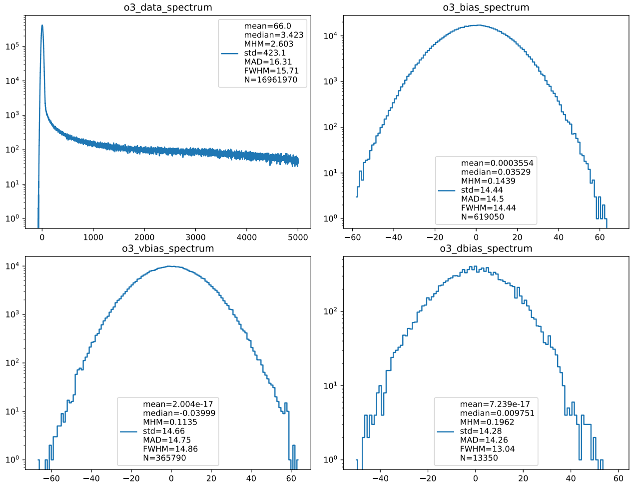
$$g^2\lambda = 33.11$$

$$g = 12.17$$

$$\lambda = 0.22$$

$E < 5000$ADU (+0 border) spectra

```
python Image.py analyse ImagePresentation/mean
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 5000.0 0.0 --plot-spectrum
```



estimations

$$\sigma = 14.4$$

$$g\lambda = 3.42$$

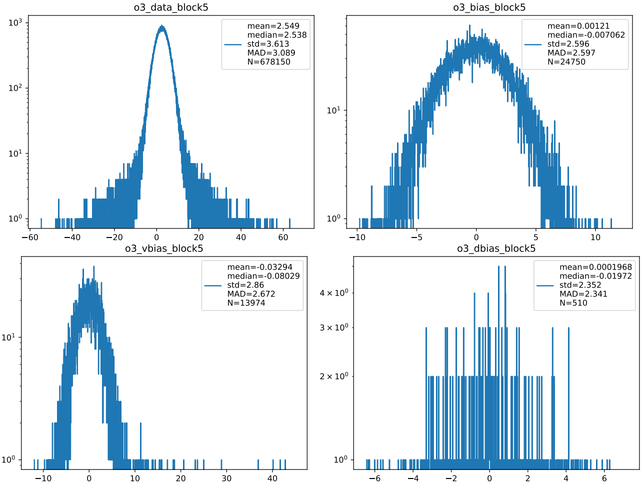
$$g^2\lambda = 55.44$$

$$g = 16.21$$

$$\lambda = 0.21$$

mean block 5x5 $E < 100$ADU $(+3)$

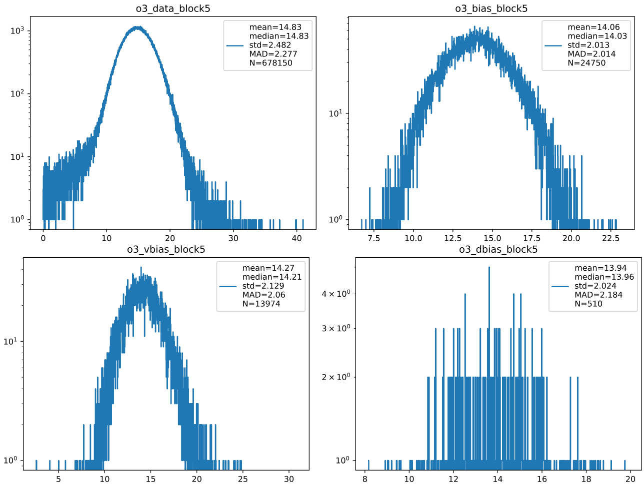
```
python Image.py analyse ImagePresentation/blockmean "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --remove-hits 100.0 3.0 --plot-block-spectrum --block-function "np.nanmean(x)"
```



$$g\lambda = 2.54$$

std block 5x5 $E < 100$ADU (+3)

```
python Image.py analyse ImagePresentation/blockstd
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-block-spectrum --block-function "(lambda y: np.nan if y==0 else
    y)(np.nanstd(x))"
```



$$\sigma = 14$$

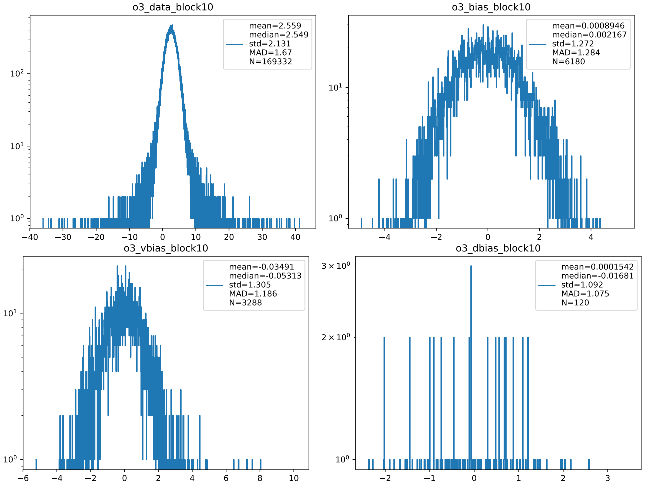
$$g^2\lambda = 23$$

$$g = 9.07$$

$$\lambda = 0.28$$

mean block 10x10 $E < 100$ADU (+3)

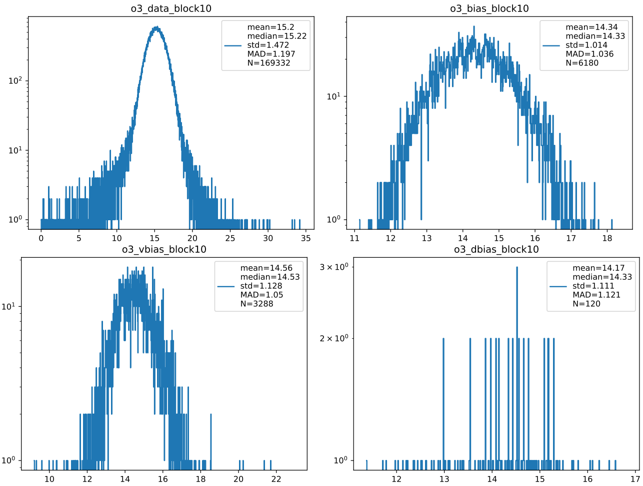
```
python Image.py analyse ImagePresentation/blockmean
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-block-spectrum --block-function "np.nanmean(x)"
```



$$g\lambda = 2.55$$

std block 10x10 $E < 100$ADU $(+3)$

```
1   python Image.py analyse ImagePresentation/blockstd
        "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
        remove-hits 100.0 3.0 --plot-block-spectrum --block-function "(lambda y: np.nan if y==0 else
        y)(np.nanstd(x))"
```



$$\sigma = 14.3$$

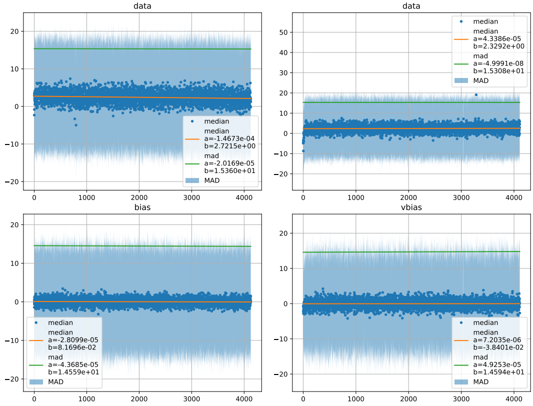
$$g^2\lambda = 27$$

$$g = 10.41$$

$$\lambda = 0.24$$

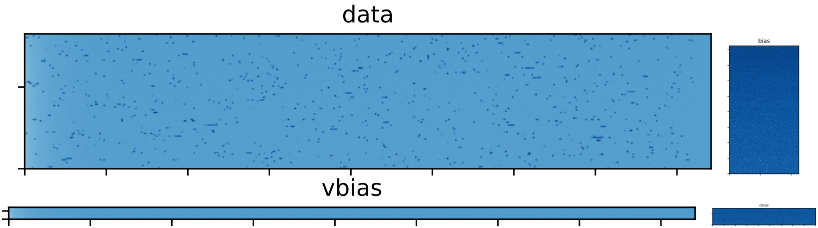
evolution of DC through time $E < 100.0$ADU (+3.0)

```
1 python Image.py analyse ImagePresentation/dcevo
    "/share/storage2/connie/data/runs/*/runID_*_03326_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-sections
```



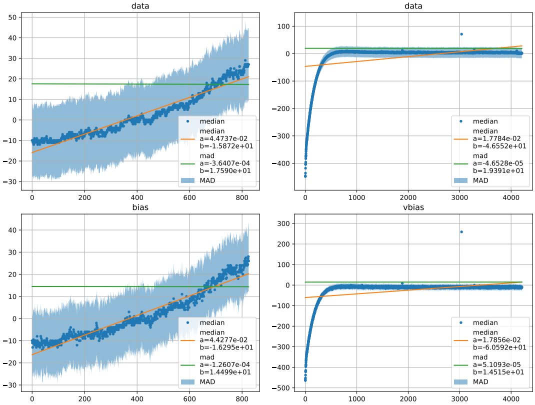
1x5

```
1 python Image.py analyse ImagePresentation/median1x5
    "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --plot-sections --
    plot-spectrum
```



projections of the raw image

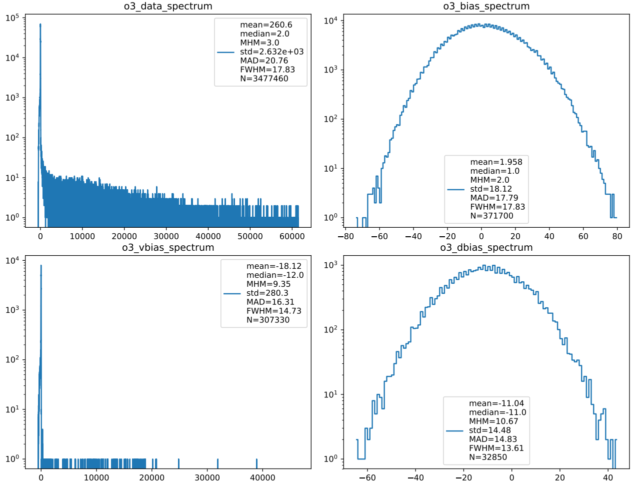
```
python Image.py analyse ImagePresentation/median1x5
    "/share/storage2/connie/data/runs/∗/runID_∗_12000_∗_p∗.fits.fz" --ohdu 3 --plot-sections --
    plot-spectrum
```



spectra of the raw image

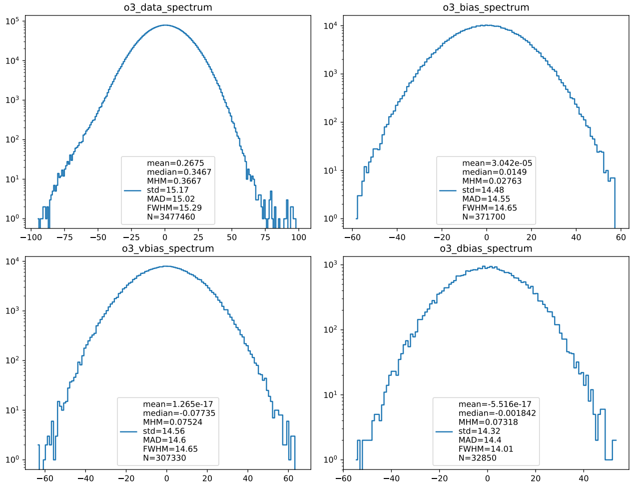
distributions are crowded with outliers

```
python Image.py analyse ImagePresentation/median1x5 "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --plot-sections --plot-spectrum
```



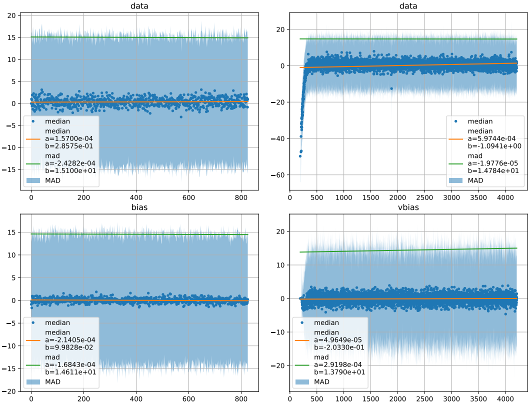
1x5 spectra with line and col corrections 1x5

```
1 python Image.py analyse ImagePresentation/dcevo1x5
    "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-sections --plot-spectrum
```



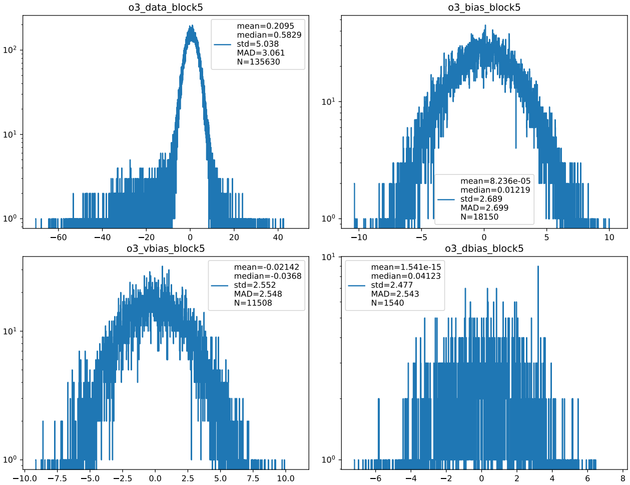
1x5 evolution of DC $E < 100.0$ADU (+3.0)

```
python Image.py analyse ImagePresentation/dcevo1x5
    "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-sections --plot-spectrum
```



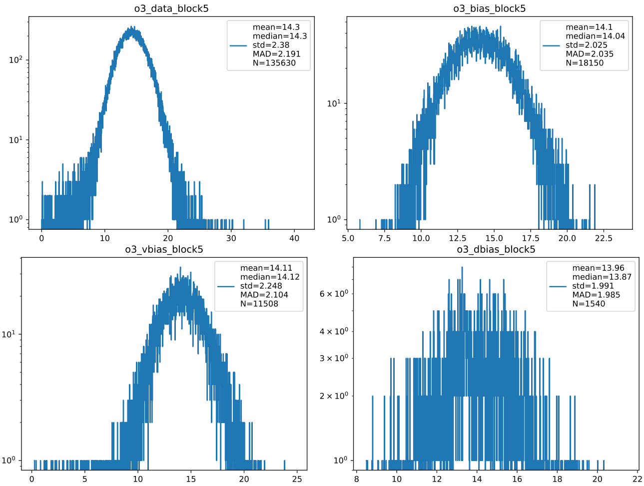
1x5 block 5x5 $E < 100.0$ADU (+3.0)

```
1   python Image.py analyse ImagePresentation/blockmean1x5
        "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --params-mode mean --
        remove-hits 100.0 3.0 --plot-block-spectrum --block-function "np.nanmean(x)"
```



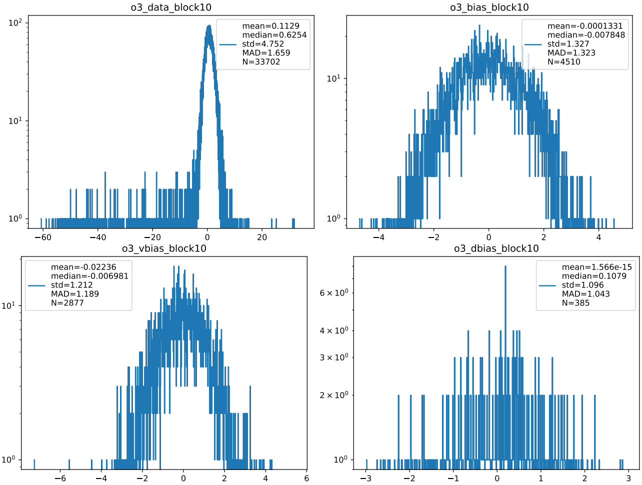
1x5 block 5x5 $E < 100.0$ADU $(+3.0)$

```
python Image.py analyse ImagePresentation/blockstd1x5 "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --params-mode mean --remove-hits 100.0 3.0 --plot-block-spectrum --block-function "(lambda y: np.nan if y==0 else y)(np.nanstd(x))"
```



1x5 block 10x10 $E < 100.0$ADU (+3.0)

```
python Image.py analyse ImagePresentation/blockmean1x5
    "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-block-spectrum --block-function "np.nanmean(x)"
```



1x5 block 10x10 $E < 100.0$ADU (+3.0)

```
python Image.py analyse ImagePresentation/blockstd1x5
    "/share/storage2/connie/data/runs/*/runID_*_12000_*_p*.fits.fz" --ohdu 3 --params-mode mean --
    remove-hits 100.0 3.0 --plot-block-spectrum --block-function "(lambda y: np.nan if y==0 else
    y)(np.nanstd(x))"
```