# Exercise 2: Basic Processing of Imaging Data

*Astronomical Data Processing Course, 2022, Copenhagen University,*
*Lise Christensen, Luca Izzo, Christa Gall, Marianne Vestergaard*

### Learning outcomes

After completing this exercise, the student will be able to:

- Examine and critically evaluate the files necessary for imaging data reduction
- Perform basic imaging data processing
- Write a data reduction script
- Apply error propagation in the data reduction

#### Python modules needed for this exercise

`astropy`
`ccdproc` ( For lots of examples and information on how to solve the questions in this exercise see https://ccdproc.readthedocs.io/en/latest/)

### Background reading material:

A good background source for describing the processing of CCD data can be found here: https://www.astropy.org/ccd-reduction-and-photometry-guide/v/dev/notebooks/00-00-Preface.html

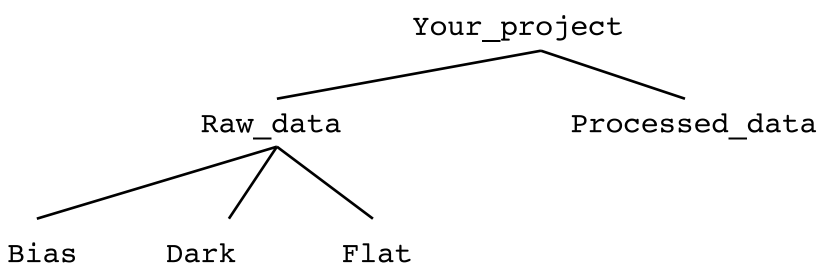
The Flat Sky: Calibration and Background Uniformity in Wide Field Astronomical Images, F. Chromey, PASP, 1996, 108 , 94

Cosmic ray hit rejection algorithm, LA cosmic, by P. van Dokkum, PASP 113, 1420 (2001).
P. van Dokkums web site.: http://www.astro.yale.edu/dokkum/lacosmic/

---

### Be organized: Maintain a clear overview of the files

It is a good idea to keep track of all your files you have and will create. For example, you can choose to have separate directories with raw and reduced data:



**Clean up**
As you process the data you will also create temporary files. They are good for checking the quality of the processing. But as you progress with the data reduction you will not need all the temporary files. Unneeded files clutter and take up disk space. So, you need to evaluate which files to keep and which to delete. PS: when you write scripts, some files are easy to recreate if you happen to need them again later.

---

## *Image Data Processing – an overview*

**A) Image Processing Overview**

The CCD image is a matrix of numbers – X and Y values are the pixel positions, the Z value is the intensity or brightness in the pixels. When a CCD is exposed, each pixel collects a certain number of photons that are converted to electrons. This is the signal, which follows poison-statistics. When the CCD is read out the signal is slightly changed – some changes are deliberate (e.g., bias) and some are not (e.g., dark current, read-out noise). Details can be found in Steve Howell's book on `CCD Astronomy' chapters 2 and 3.

The goal of data processing is to correct the raw read-out signal in the image for the deliberate changes made and the flaws of the CCDs in an attempt to reconstruct the original signal. These corrections come in two types: additive and multiplicative. As each image has noise and each correction is, after all, an estimate of the true correction, each processing step will add noise and uncertainties. *An important goal is to minimize the addition of extra noise and systematics during the data processing.*

Briefly, the needed calibration images are:

| | |
|---|---|
| Bias | The electronically added offset (DC level) – to be subtracted. For some CCDs this is a constant offset, while others have spatial structure and a bias frame is needed. A zero seconds exposure with closed shutter. Level ~100 – 1000 ADU (analogue to digital units) per pixel. |
| Dark current | The thermal signal intrinsic in the CCD, scaled to the actual exposure time – to be subtracted. Can be constant or have spatial structure, like the bias. Dark current increases with exposure time. An exposure with EXPTIME > 0 and closed shutter. Level for modern CCDs: ~2 – 10 electrons/pixel/hour. |
| Shutter | Correction for the difference between the set exposure time and the actual. Important for short (~1 sec) exposures only. Not done for these exercises. |
| Fat zero | Some columns ``lag behind": there is always an offset between the level in these columns and in the rest of the image. Are there any present in your data? Corrections are not needed. |
| Flat field | The response of the CCD pixels is not the same for all pixels. It depends on position and spectral distribution of the incoming light. A flat field is an image of an evenly illuminated surface – imaginary for sky (twilight) flats. Flats are different depending on the light source and filter band pass used. Correction is multiplicative based on a normalized flat. Optimal level: 10,000-25,000 ADUs. |

The basic data processing with application of the additive and multiplicative corrections is (omitting shutter and fat zero):

```
                    (raw_image – bias – [dark current])
final_image =    -------------------------------------------
                  normalized(flat – bias – [dark current])
```

---

**B) Statistics and Sources of Error**

To get realistic estimates of the uncertainties, the noise in the images must be known. Sources of errors include:

| | |
|---|---|
| Read-out noise (RON) | The noise inherent to the electronics reading out the CCD. New CCDs have lower RON of a few electrons per pixel per read out. The RON can be determined as the standard deviation in the difference between two bias frames when certain conditions are fulfilled. |
| Photon noise | Statistical `shot noise' in the number of detected photons, governed by Poisson statistics (see separate study note by Peter Jakobsen). This means that the variance of a Poisson stochastic variable equals its mean and that the statistical error on the signal in each pixel due to photon shot noise can be estimated as the square root of the number of detected photons for a given exposure. |
| Background subtraction | The uncertainty in the determination of the sky background. |
| Cosmic Ray hits (CRs) | High-energy particles that pass through the detector and deposit large amounts of energy, mimicking the deposition of energy by which CCDs detect photons. As CRs are single events that do not land on the same part of the CCD in subsequent exposures, they are best located by comparing multiple images. |
| Hot/Cold columns | Hot columns have particularly sensitive pixels; intensity level is always (somewhat) higher compared to the rest of the image. Cold columns have rather insensitive or `dead' pixels.<br>Direct re-scaling of the intensity rarely works. These pixels cannot be corrected and need to be ignored. Avoid placing important objects on these artifacts. Shifting the pointing between exposures can help. |

The intensity level in raw images is in ADU (analogue-to-digital units). To get the number of photons, you need the conversion factor or gain, which gives the number of electrons per ADU. It can be computed from the ratio of two flat fields in the same filter. Ideally, this ratio contains only noise and if the RON is known, the gain can be estimated.

---

**C) Combining calibration Images**

To limit adding noise when you calibrate your data, you need calibration images with high signal-to-noise (i.e., 'good statistics'). You do this by combining *many* single exposures, i.e. making a stack of identical images, and on a pixel-by-pixel case taking the average or median value. In this process, pixels containing signal from CR hits should be excluded too. The pixel-to-pixel variations in the bias and dark images are due to noise only. Large-scale gradients in biases and darks are important to correct for.

Generally, the more frames you combine, the lower the resultant noise. See Massey's 1997 write-up on 'How many and what calibration frames do you need" for more information. At a minimum to get a good S/N:

Bias - 10 or more, median filter or median combine
Dark - 3 or more 1-hour exposures, median filter/combine
Flats - 10 or more exposures with levels of ~20,000 $e^-$ /pixel

Consider when a normal average is not a good idea? A median-combine is another option. Both average and median combination may use some of the rejection algorithms. With a minimum of 3 images (or odd-multiples) you can eliminate almost all CR hits. Particularly, long dark exposures will contain many CRs. You should use a method that works well on few images (3-5) and does not result in systematics, like too high or too low levels while still rejecting the CRs.

In the Python module `CCDproc` you can use the '`combine`' function with different choices of pixel rejections. This includes:

`sigma_clipping`
`minmax_clipping` : rejects values above and below some supplied value
`clip_extrema`

Darks: It is important to bias-subtract them and combine them according to their exposure times if this is not the same, since the dark current level scales with the exposure time.

---

**D) Process and Combine the Flats – a special note**

The flat fields of a particular filter and illumination source must be processed separately. As each flat also contains a bias level and dark current, these corrections are done first. Combining flats has one additional twist: if during the flat field exposures, the lamps illuminating the white spot on the dome flickered or faded, the multiple flats we took in each filter would have slightly different average values throughout the image. Similarly, a set of sky-flats taken under changing light conditions, and with different exposure times, invariably have different flux-levels. This will throw off the median-combine in that it will always take the pixel from the middle exposure image and thus not reduce the noise.

---

## E) Arithmetic on images

Working with 2D image arrays you sometimes need to perform simple arithmetic computations on images just as you would with numbers. In Python there is no direct way to do this with FITS images using a single command, so you need to load the fits files, and do the computation, then save the output fits file. Following short example is taken from:
https://stak-notebooks.readthedocs.io/en/latest/images.imutil.html#imarith-imdivide
Note, this document explains how to deal with divisions by '0' within image computations.

```
# First read in your data
>>> hdu1 = fits.open('data1.fits')
>>> hdu2 = fits.open('data2.fits')


# Here we add hdu2-ext1 to hdu1-ext1 by using the shortcut += operator
>>> hdu1[1].data += hdu2[1].data
```

```
# Save your new file
# The overwrite argument tells the writeto method to overwrite if file already exists
>>> hdu1.writeto('output_data.fits', overwrite=True)

# Close hdu files
>>> hdu1.close()
>>> hdu2.close()
```

When reading in the FITS files from raw data files, the data values are assigned as 16 Bit integers. So, when you perform any operations on arrays, Python may encounter some numerical problems because it retains the data type. To avoid such problems (which may not crash Python, but just give an output that is not correct!) you must convert your data to floats. One way to do this is to simply multiply your array data with 1.0 . E.g. array = array*1.0.

Alternatively, you can read in the FITS data and specify the datatype to be a float:

```
>>>image = fits.getdata(filename , dtype='f')
```

---

**F) Normalizing the Flat Fields**

A flat field should correct for pixel-to-pixel variations in the sensitivity of the pixels. It should correct the pixel values such that they each reflect the same signal for a given incident photon flux. That is, the correction is a relative correction between the pixels. Therefore, the master flat should have pixel values with a median of 1.0. As the intensity levels in flat field images are high (~20,000 ADUs) so to get good statistics, the master flat should be normalized before it is applied to science data. Here, normalization means dividing the frame by its average value.

There are different ways of choosing the normalization. It is a good idea to test different statistics to ensure you get the result you want. For images with no significant vignetting (i.e. the field is not illuminated uniformly), simply dividing every pixel in the flat field image, by the mode of the image works well. What might be a good solution if the flat has vignetting?

---

**G) Aligning Multiple Images for Stacking**

Often we need to build up signal to noise using deep (i.e., long) exposures. Making extremely long exposures is not advisable due to imperfect telescope tracking, change in focus/seeing or weather conditions, limited length of nights, potential failure of instrument, source saturation, and/or the collection of a large number of CR hits, to name a few issues. The solution is to obtain several shorter exposures and combine them.

For science exposures alignment is needed since the images are typically offset slightly in position from one another. To measure the offsets, one needs to measuring position(s) of star(s) common to all images. Alternatively, it is possible to use the FITS header keywords that supply the pointing of the telescope/instrument at the time of observations through the World Coordinate System (WCS), but this will only work when the astrometric solution is accurate. In the Python module CCDProc there is a code called wcs_project that will use the WCS information to align a new image to a reference image.

When combining images, you also need to propagate the associated errors. The corresponding variance is the sum of the variance of the individual images divided by the number of non-rejected pixels, *N.* For example, using the type of combination 'average', `minmax rejection, and the k-sigma clipping methods, the stacked frame variance in the final variance image is:

$$var(stack) = \frac{1}{N^2}\sum_{i}^{N}\sigma_i^2$$

When the images are median combined (see the separate write-up on using the median), the variance is

$$var(median - stack) = f(N)\frac{1}{N^2}\sum_{i}^{N}\sigma_i^2$$

where *f(N)* is 1 for $N \leq 2$, and it converges toward $\sqrt{\pi/2}$ for $N \rightarrow \infty$ .

This is approximately also valid for a Poisson stochastic variable like photon flux because a sum of Poisson distributions quickly approaches a normal distribution (see separate study note: ``AD_Noise-PJnote.pdf'' (Day 5 Digitization of noise)).

---

**H) Error Propagation**

Any scientific measurement involves error analysis. This is the only way to quantify the significance of our detections. To evaluate the uncertainties in our final, calibrated science image it is convenient to generate the error image. This is an estimate of the errors in each pixel accumulated during the data processing. The same error statistics are at play as when we make individual measurements in the laboratory in Physics classes. Here, we simply have a matrix or grid of measurements, with each pixel value having an error associated. However, not all errors are Gaussian, like the read-noise RON. Photon statistics follow Poisson statistics (You may find this link useful: [http://www.umass.edu/wsp/statistics/lessons/poisson/index.html](http://www.umass.edu/wsp/statistics/lessons/poisson/index.html) ). We recommend that you read the supplementary note on this by Peter Jakobsen; it's on Absalon.

Since it is the photons that obey Poisson statistics, and due to simplicity, we choose to work in units of electrons. The signal in the raw CCD image is often in units of counts (ADU) and needs to be converted to number of photons (i.e., electrons) with the appropriate gain *g* (in units of electrons/ADU): *raw(e) = raw(ADU)* × *g*. Then, according to Poisson statistics the uncertainty in the Poisson signal is simply the square root of the signal: $Noise = \sqrt{S}$. It is therefore convenient to work with the variance: $var = \sigma^2 = Noise^2 = S$. The starting point in creating our variance image is the raw image in units of electrons, *raw(e),* as noted above. In the following, all images, RONs, and error contributions are in units of electrons.

For the noise image (= $\sqrt{\text{variance image}}$ ) to represent the uncertainty in the final science image we need to propagate all the error contributions (in units of electrons) involved in the data processing using the standard error statistics and propagation methods, as outlined below.

Next, we will go through each of the steps in the data reduction process and the relevant error contributions. It is very general, and we are strict in including all the terms. It is very illustrative to look at your data to evaluate which contributions matter and which are insignificant and why. Also, if you evaluate that there is not any significant dark current in your data, then you just set *var(dark) = 0* in the equations below.

1. ***Raw, uncorrected images:***

The raw image `*raw(e)'* contains photons and read-noise (plus a bias level from the electronics that we need to subtract to get the pure photon-signal). This means that the variance of a raw image *signal* is:

$$var(raw) = \sigma_{raw}^2 = S_{raw} + RON^2 = (raw\ science\ image - bias) + RON^2 \qquad \text{[eqn. 1]}$$

**2. *Bias-corrected images, $S_{bc}$:***

Since the raw image `*raw(e)'* contains the electronic DC level, you need to first subtract the bias level from any image: $\boldsymbol{S_{bc} = (raw - bias)}$. The bias level may be constant or a `master-bias'. In case of using a master-bias (MB), we need to compute the errors contributed by generating this mean (or median) combined frame of $N_{bias}$ individual bias frames: $MB = \sum_i^N bias_i / N_{bias}$ . We propagate the errors on the bias-corrected image using the standard propagation equation (`ophobningsloven' in Danish):

$$var(S_{bc}) = \sigma_S^2 = \left(\frac{\partial S_{bc}}{\partial raw}\right)^2 \sigma_{raw}^2 + \left(\frac{\partial S_{bc}}{\partial bias}\right)^2 \sigma_{bias}^2 = \sigma_{raw}^2 + (-1)^2 \sigma_{bias}^2$$

The master-bias will contain a noise contribution from the uncertainty in determining the bias level (or DC-level), σ(DC), *and* a contribution from the reduced RON level obtained by combining $N_{bias}$ images: RON/$\sqrt{(number\ of\ bias\ images)}$:

$$var(MB) = var\left(\frac{\sum_{i=1}^N bias_i}{N_{bias}}\right) = \frac{1}{N_{bias}^2} var\left(\sum_i^N bias_i\right) \cong \frac{1}{N_{bias}^2}\left[\sum_i^N (RON^2)_i\right] = \frac{RON^2}{N_{bias}},$$
$$\text{such that } \sigma^2(bias) = \sigma^2(DC) + \left(\frac{RON^2}{N_{bias}}\right) K(N_{bias})$$

Here we use the assumption that RON is the same for each of the $N_{bias}$ bias frames. *K(N)* is a factor that depends on the number of images combined and the combination method. K(N)=1 when you combine your frames with a geometric average. When you median combine your images, *K(N)* is 1 for $N \leq 2$, and it converges toward $\sqrt{\pi/2}$ for $N \to \infty$ (see the notes on statistics of mean and median on Absalon).

How is σ(DC) estimated? The counting error is not realistic as the DC level is not a real photon signal. If the CCD is well behaved, the DC level is constant in time and well-determined ($\sigma^2(DC) \approx 0$). But consider this: If you have mis-determined the DC level by, say, 1 ADU, then this is a *systematic error*, different from the stochastic error we are characterizing here. When you do analysis of the science images, you will always subtract the sky-level background from your target signal, and this systematic error per pixel will be eliminated (*Phew!*). However, you should still be careful to determine the bias level well, as the later flat fielding process will also scale any systematic offset. So you have good reasons to use many bias frames to get the DC level accurately determined and to test that the CCD bias level is stable. If the DC level varies in time and you do not have an overscan or prescan region to check the overall bias level, you have another serious problem that is not easily accounted for with the variance image. In the current case, you can safely set σ(DC)=0:

$$\sigma^2(bias) \approx \left(\frac{RON^2}{N_{bias}}\right) K(N_{bias}) \qquad \text{[eqn. 2]}$$

***3. Dark corrected Images***

In addition to the bias level, our images contain dark current, which we correct for by subtracting the dark-current level accumulated during the exposure. We typically obtain several, hour-long dark exposures in order to get good statistics when we determine the dark-current level in units of electrons per pixel per hour [e/pix/hr]. Since every frame read out with our detector contains a bias (DC) level [and read-noise] we need to correct our dark images for the bias-level, using our master-bias. Thereafter, we (average or median) combine the individual dark frames and scale to the exposure time, $T_{EXP\text{-}OBJ}$, of the frame that we intend to correct. This can be a science exposure or a flat field. We will call that frame 'OBJ'.

### <u>*3.1 A Single Dark Frame:*</u>

Let us first look at the variance of a single, bias-corrected dark-frame:

***dark = (raw-dark – bias)**:*

$$var(dark) = var(rawdark) + var(bias) = (rawdark - bias) + RON^2 + \sigma^2(bias) \quad \text{[eqn. 3.1]}$$

where $\sigma^2(bias) = \frac{K(N_{bias})RON^2}{N_{bias}}$, as derived in equation 2 above.

For a single, bias-corrected dark-frame, <u>*scaled*</u> to the 'OBJ' exposure time:

$$dark(T_{EXP-OBJ}) = (dark(T_{EXP-DARK}) - bias) * \left[\frac{T_{EXP-OBJ}}{T_{EXP-DARK}}\right],$$

the variance is [here, we use: var(constant*S) = constant$^2$ * var(S)]:

$$var\left(dark(T_{EXP-OBJ})\right) = var(dark)\left[\frac{T_{EXP-OBJ}}{T_{EXP-DARK}}\right]^2, \quad \text{[eqn. 3.2]}$$

with *var(dark)* as determined in eqn. (3.1).

### <u>*3.2 Multiple Dark Frames:*</u>

Now, let's stack (i.e., mean or median combine) a number, $N_{dark}$ of individual, bias-subtracted dark-frames, called $D_i$ (no scaling yet), and the combined frame being *<D>* :

$$< D(T_{EXP-DARK}) \geq \frac{1}{N_{dark}}\sum_{i=1}^{N_{dark}}(dark(T_{EXP-DARK}) - bias) = \frac{1}{N_{dark}}\sum_{i=1}^{N_{dark}} D_i$$

The variance of *<D>* is:

$$\begin{aligned} var(< D >) = var\left(\frac{\sum_{i=1}^{N_{dark}} D_i}{N_{dark}}\right) &= \frac{K(N_{dark})}{N_{dark}^2} var\left(\sum_{i=1}^{N_{dark}}(D_i)\right) = \frac{K(N_{dark})}{N_{dark}^2}\sum_{i=1}^{N_{dark}} var(D_i) \\ &= \frac{K(N_{dark})}{N_{dark}^2}\sum_{i=1}^{N_{dark}}\left(RON^2 + D_i + \sigma^2(bias)\right) \\ &= K(N_{dark})\left(\left(\frac{RON^2}{N_{dark}}\right) + \frac{\sum_{i=1}^{N_{dark}} D_i}{N_{dark}^2} + \left(\frac{RON^2}{N_{dark}N_{bias}}\right) K(N_{bias})\right) \\ &= K(N_{dark})\left(\left(\frac{RON^2}{N_{dark}}\right) + \frac{< D >}{N_{dark}} + \left(\frac{RON^2}{N_{dark}N_{bias}}\right) K(N_{bias})\right) \quad \text{[eqn. 3.3]} \end{aligned}$$

Here we use that for independent $D_i$ measurements, the variance of the sum is the sum of the variances. Again, *K(N)* is the `penalty' of our combination method (see subsection 2 above – or section G). When we scale this stack of $N_{dark}$ individual, bias-subtracted dark-frames so to create a master-dark (MD):

$$MD(T_{EXP-OBJ}) = \left[\frac{1}{N_{dark}}\sum_{i=1}^{N_{dark}}(dark(T_{EXP-DARK}) - bias)\right]\left[\frac{T_{EXP-OBJ}}{T_{EXP-DARK}}\right] = <D>\left[\frac{T_{EXP-OBJ}}{T_{EXP-DARK}}\right]$$

The variance of this averaged or median combined stack is:

$$var(MD) = \sigma_{MD}^2 = var(< D >)\left[\frac{T_{EXP-OBJ}}{T_{EXP-DARK}}\right]^2, \qquad \text{[eqn. 3.4]}$$

with *var(<D>)* as defined in eqn. (3.3). The noise contribution from the dark correction is thus the signal, the reduced RON in the combined dark, and a similar contribution from the reduced RON from the master-bias.

*3.3 Science frame – bias and dark corrected, $S_{bdc}$:*

As a result, a bias and dark-subtracted image-frame: ***$S_{bdc}$* = (raw – bias – dark)** will have the variance:

$$var(S_{bdc}) = var(raw) + \sigma^2(bias) + \sigma^2(dark)$$

$$= \left((raw - bias) + RON^2 + \left(\frac{RON^2}{N_{bias}}\right) K(N_{bias}) \ + \ var(MD)\right) \qquad \text{[eqn. 3.5]}$$

where the last line shows the contributions from a master-bias (eqn. 2) and master-dark (you will need to insert the *var(MD)* contribution from equation 3.4). Recall that we need the photon-signal from the \`raw' image, hence you need to bias-subtract the raw data frame. It is clear that since each frame (calibration or science) needs to be bias-corrected, we have a strong interest in keeping the noise in the master-bias low by combining a large number of individual bias frames.

## *4. Flat fielded Images*

The flat field image itself contains the illumination signal, a bias level, and dark current. The latter two are subtracted so to isolate the illumination signal of interest: ***F = (rawF – bias – dark).*** The signal in the master flat-field *f* is the averaged collection of flat fields *<F>,* normalized by the average (or median; as a sum of Poisson distributions quickly approaches a normal distribution, the formula still holds approximately) overall flux level $\bar{F}$ in the flat field image: $\boldsymbol{f} = \frac{<F>}{\bar{F}}$. Here, **<F>** = $\frac{\sum_{i=1}^{N_{flat}} F_i}{N_{flat}}$, where *i = 1, ..,$N_{flat}$* indicates the individual flat-field images entering the stacked, average flat field.

*4.1 Flat fields – ignoring bias and darks:*

If we temporarily ignore the noise-contributions from the bias and dark corrections, it is easier to see how the variance propagates when we stack and normalize. The variance in a single flat field is: $\boldsymbol{\sigma_f^2}(\mathbf{1\ flat}) = (\boldsymbol{RON^2} + \boldsymbol{F})$**,** where *F* is the flat image with the (preferably high) signal in each pixel. When we stack *N* flat fields and average these flats, the variance on the averaged flat is (\`variance in the mean'):

$$var(mean\,flat, signal\ per\ pixel) = var(<F>) \ = var\left(\frac{\sum_{i=1}^{N} F_i}{N}\right) = \frac{1}{N^2} var \sum_i^N F_i$$

$$= \frac{1}{N^2}\sum_i^N var(F) = \frac{1}{N^2}\sum_i^N (RON^2 + F)_i = \frac{RON^2}{N} + \frac{1}{N^2}\sum_i^N F_i = \frac{RON^2}{N} + \frac{<F>}{N}$$

Again we use: variance(sum) = sum(variances) since the $F_i$ are independent measurements. Thus,

$$\sigma_f^2(N\ stacked\ flats) = \left(\frac{RON^2 + < F >}{N_{flats}}\right) \qquad \text{[eqn. 4.1]}$$

Then, when we normalize the averaged stack with the average flux level (or an image with a locally smoothed level) $\bar{F}$ in units of electrons, the variance of the normalized flat *f* is:

$$\sigma_f^2(N\ stacked\ normalized\ flats) = \frac{1}{\bar{F}^2}\left(\frac{RON^2 + < F >}{N_{flats}}\right) = \frac{RON^2}{N_{flats}\bar{F}^2} + \frac{f}{N_{flats}\bar{F}} \qquad \text{[eqn. 4.2]}$$

As noted in sections 2 and 3, when the stack is median-combined, the variance includes a `penalty' factor *K(N)* $\rightarrow \sqrt{\pi/2}$ for *N* $\rightarrow \infty$ . Section F and the extra write-up on using the median in astronomical analysis have more information.

*4.2 Flat fields – bias and dark corrected:*

As any other image frame obtained at the telescope, flat fields also contain a bias level and dark-current that scales with the exposure time of the flat fields. In the following we propagate the errors from our standard bias and dark-corrections of the flat fields.

*4.2.1 A Single flat field:*

A single master-bias and master-dark (MD) corrected flat field is: $F = rawF - MB - MD(T_{EXP\text{-}FLAT})$, where the MD is scaled to the exposure time of the flat . The variance is:

$$var(F) = var(rawF) + var(MB) + var[MD(T_{EXP-FLAT})]$$

Inserting the relevant terms from this section and equations (2) and (3.4), we get

$$var(F) = RON^2 + F' + \left(\frac{RON^2}{N_{bias}}\right) K(N_{bias}) + var(< D >)\left[\frac{T_{EXP-FLAT}}{T_{EXP-DARK}}\right]^2$$

with *var(<D>)* given by equation (3.3) and *F'* is the bias-corrected flatfield frame (i.e., the unprocessed *F* frame minus the bias-level) as we need the `raw' photon signal here.

*4.2.2. Multiple flat fields:*

If we stack a number $N_{flat}$ of such images, the variance of this resultant frame *<F>*, is

$$var(stacked, corrected\ flats) = var(< F >) = \frac{K(N_{flat})}{N_{flat}^2}\left(\sum_i^N var(F)_i\right)$$

$$= \frac{K(N_{flat})}{N_{flat}^2}\sum_i^N \left(RON^2 + F' + \left(\frac{RON^2}{N_{bias}}\right) K(N_{bias}) + var(< D >)\left[\frac{T_{EXP-FLAT}}{T_{EXP-DARK}}\right]^2\ \right)_i$$

$$= K(N_{flat})\left\{\frac{RON^2}{N_{flat}} + \frac{<F>}{N_{flat}} + \frac{RON^2 K(N_{bias})}{N_{flat}N_{bias}} + \frac{K(N_{dark})}{N_{flat}^2}\left[\frac{RON^2}{N_{dark}} + \frac{< D >}{N_{dark}} + \frac{RON^2 K(N_{bias})}{N_{dark}N_{bias}}\right]\left[\frac{T_{EXP-FLAT}}{T_{EXP-DARK}}\right]^2\right\} \qquad \text{[eqn 4.3]}$$

Finally, the normalized stack of flat fields (the master-flat) has variance:

$$\sigma_f^2(N\ stacked\ normalized\ flats) = var\left(\frac{<F>}{\bar{F}}\right) = \frac{var(<F>)}{\bar{F}^2} \qquad \text{[eqn 4.4]}$$

with *var(<F>)* defined in eqn. (4.3). With the various terms in eqns. (4.3) and (4.4), it is clear that with a low dark-current that is then not corrected for, a large signal $\bar{F}$ in the flat fields, as well as a combining a large number of biases and flats helps to keep the noise-contributions low.

**Ignoring terms?** Recall, that you can always evaluate the individual noise contributions (i.e., the terms in these equations) and determine which terms may be insignificant and thus ignored. For example, for our data with no dark-current correction: $\sigma_{dark}^2 = 0.$ With high signal in the flat ($\bar{F}$ large) and a large number of flats and bias frames (the product $N_{flat}N_{bias}$ is very large), then the third term in eqn. (4.3) is insignificant and eqn. (4.4) reduces to (here we also assume $\sigma_{dark}^2 = 0$):

$$\sigma_{flat}^2(masterflat) = \frac{K(N_{flat})}{\bar{F}^2}\left\{\frac{RON^2}{N_{flat}} + \frac{<F>}{N_{flat}}\right\} = K(N_{flat})\left\{\frac{RON^2}{N_{flats}\bar{F}^2} + \frac{f}{N_{flats}\bar{F}}\right\} \qquad \text{[eqn 4.5]}$$

That is, the variance reduces to equation (4.2) with the `penalty factor' *K(N)* explicit here.

## *5. Science Images: Bias, Dark, and Flat Corrected frames*

The processed (i.e., flat fielded) science image is ***PS = (raw image – bias – dark)/f*** (in units of electrons). Again, we propagate the errors on the flat-fielded science image using the standard propagation equation (`ophobningsloven' in Danish):

$$var(PS) = \sigma_{PS}^2 = \left(\frac{\partial PS}{\partial raw}\right)^2 \sigma_{raw}^2 + \left(\frac{\partial PS}{\partial bias}\right)^2 \sigma_{bias}^2 + \left(\frac{\partial PS}{\partial dark}\right)^2 \sigma_{dark}^2 + \left(\frac{\partial PS}{\partial flat}\right)^2 \sigma_{flat}^2$$

$$= \left(\frac{1}{f}\right)^2 \sigma_{raw}^2 + \left(\frac{-1}{f}\right)^2 \sigma_{bias}^2 + \left(\frac{-1}{f}\right)^2 \sigma_{dark}^2 + \left(\frac{-PS}{f}\right)^2 \sigma_{flat}^2$$

$$= \frac{[S_{raw} + RON^2]_{raw} + \sigma_{bias}^2 + \sigma_{dark}^2 + PS^2\sigma_{flat}^2}{f^2}, \quad \text{where } \sigma_{raw}^2 = S_{raw} + RON^2.$$

Recall, $S_{raw}$ is the initial signal, that is, the raw data image minus the bias level (or bias-frame). This equation for *var(PS)* shows that we need to add each of the noise contributions, including $\sigma_{flat} = \sigma_f$ (the last term). In addition, there is the multiplicative correction equivalent to flat-fielding the variance image two times.

Use the definition of variance to verify that when you scale the fluxes by factor *F*, the variance is then to be scaled by $F^2$. That is, this multiplicative scaling makes sense.

Thus, to generate the resultant variance image of a flat-fielded science image, start with the raw image (in units of electrons) and add each noise contribution (for each pixel) according to standard error propagation with the bias and dark contributions as shown earlier:

$$var = \frac{\left(S_{raw} + RON^2 + \sigma^2(bias) + \sigma^2(dark) + PS^2\sigma_{flat}^2 + \sigma_G^2\right)}{f^2} \qquad \text{[eqn 5]}$$

Each contribution is that shown above (eqns. 2, 3.4, and 4.5) where the variance contribution from the dark current is scaled to the exposure time of the science image (`$T_{EXP}$(obj)' above).

The extra term $\sigma_G$ is needed when the gain is very high (above ca. 5-10 e-/ADU). This is the error introduced by the *digitization noise* within the A/D converter. Steve Howell (sections 3.8 and 4.4 in ``Handbook of CCD Astronomy'') describes this as: $\sigma_G = Gain * \sigma_{df}$, where $\sigma_{df} \approx 0.289$. Howell calls this `$\sigma'_f$. See the separate write-up ("Noise due to Digitization Errors in A-to-D Converters" by P. Jakobsen) summarizing the derivation of the digitization noise for two different types of A-to-D converters. For a converter that rounds the scaled pixel charge to the nearest integer output value the digitization noise $\sigma_G$ *is* $\frac{1}{\sqrt{12}}$ times the gain. You can ignore the digitization noise here (and for most modern data where the gain is low).

*Note:* Strictly speaking, if the digitization noise is relevant, we need to add a contribution from this noise to each of the variance contributions of equation 5. In practice, it would be relevant to substitute `$\mathrm{RON}^2$' with `$\mathrm{RON}^2 + \sigma_G^2$' (except: the $\sigma_G^2$ term already in equation 5 is that `belonging' to the $\mathrm{RON}^2$ term visible there so there is no need to substitute that particular $\mathrm{RON}^2$ term).

**Evaluating the important terms:** When you insert the full equations 2, 3.4, and 4.5 in equation 5, you have a lot of terms. Recall that you can evaluate each term to see if any of them are insignificant relative to the other contributions, such that you can ignore such terms. For example, how large is term 3 in eqn. 4.5 for your data relative to terms 1 and 2? Equation 5 (with all the terms inserted) is good for evaluating how to optimize your observing run.

**Important:** Note that the second to last term in the above equation, $PS^2\left(\frac{\sigma_{flat}}{f}\right)^2$, contains the relative error in the master flat field, $\sigma_{flat}/f$ . So if the flat field has a 5% error, this will propagate into your science image uncertainty and you will never be able to do 1% photometry! This emphasizes the quality requirements of your master flat field for the photometric accuracy that you aim for.

---

## *Exercises: Basic Imaging Data Processing*

We will use data of an open stellar cluster, M67, obtained with the EFOSC instrument mounted on the ESO 3.6m telescope on La Silla, Chile. The detector size for the EFOSC CCD is 2048x2048 pixels, which can be read out in binned or un-binned mode, in a fast or normal mode.

Information on where the data is found is in the document "Where to find the data for course – where to work on the data" on Absalon. The data is split in two separate directories: one containing all the science data, and one with all the calibration data. Note that it is not all the files in the calibration directory that will be useful for calibrating the data.

### *Exercise 2.1: The Bias and Dark Images*

1. Inspect the information from the FITS headers:
   List the exposure time of all science frames. Find the dimensions of the science images (in the SCIENCE directory from the tar file), i.e., the number of pixels in the X and Y directions. Find all the equivalent bias frames in the CALIB data directory, e.g. using the `dfitspy` function).
   Make a file list of the relevant bias images that can be used.

2. Examine a few individual bias frames, picked at random, and display them with DS9, and using appropriate dynamic ranges on your image display. Does the bias level look completely uniform over the entire 2D array, or can you see strange features?
   Is the entire detector field useful when determining statistical properties?

3. **Script.** Write a small script that loads the FITS files of the bias images and determine pixel statistics of selected regions on the 2D frame :
   What is the mean bias level ?
   What is the standard deviation ?
   Are there changes across the full images ?
   Does the level change between frames?

   Write the results in a table for your report.

4. **Read out noise (** RON): Explain why bias images can often be used to determine the RON? When will it not work?
   Measure the RON from a single bias image. Plot a histogram of the data values. RON can be given in electrons, or in ADU. The conversion factor (gain) is written in the FITS headers. Most times, so is the RON, but this is not the case here. Compare the RON estimated above with that determined from Howell's equation on page 73. What is the difference? Why do we prefer the latter estimate?

5. **Combined data.** Combine a number of bias files into a master-bias fits file using the `CCDproc combine container` with various choices of pixel rejection types: `minmax,` and `sigclip`, and choices of how many 'sigma' outliers to exclude in the combination.

   For examples and hints you can examine section 2.3 in the reference material here:
   https://www.astropy.org/ccd-reduction-and-photometry-guide/v/dev/notebooks/02-04-Combine-bias-images-to-make-master.html

   Note that 'imagetyp= BIAS' is not a FITS header in these EFOSC data, so you cannot use this selection directly in the data we use here. Instead, you can make new sub-directories in your CALIB directory, or you can read in the bias.list file.

6. **Master bias.** Create two master-bias frames from 3 and N individual frames, respectively, with N (>20) being the maximum number of (useable) bias frames. Measure the noise (= standard-deviation) in the master-bias frames, again avoid using the edges of the detector which may have pixels that are noisier than the average.

   Determine how that noise changes when you combine a larger number of original bias frames for the master frame.

   Is the master-bias flat and free of any structure? If not, how can we evaluate whether to use this frame or a simple constant level when correcting for the bias?

7. **Overscan regions.** Estimate the bias level in the overscan regions in the master bias frame. The overscan region can be hard to see visually, but the leftmost 6 columns (when displayed in ds9) contain the overscan region, and the upper 6 rows (in the display in ds9) also contain an overscan region in these EFOSC files we work with here. Compare the level in these overscan regions with the bias level in other regions in the master bias file.

8. **Dark frames**. Locate the dark frames among the Calibration files and look at them in ds9. What sources of noise and features are present in the dark frames?

9. Write a script so subtract the bias level from the darks. Why is this necessary? It is important to pay attention to the exposure time of the darks. Why? Generate a master dark. Is it flat and structure-less and free of cosmic ray hits? Measure the dark current of the detector in units of e-/pixel/hour. At which exposure time does the dark current become similar to the RON? Do you need to account for the dark current in the reduction of your science images? Justify your reasoning.

### *Exercise 2.2: The Flat Field Images*

To ease the workload for this course, your working group may choose to split the processing of images in the 3 different filters used, and you only need to work on the analysis of one of the filters each. For the final report you must collect all the relevant measurements and discussions from you group.

1. The FITS header keyword reflecting the filter that is used in each image is called "HIERARCH ESO INS FILT1 NAME" in the data we work with here. This value can be either V, R or B filters in our case. Each group member should choose a different filter to work on.

2. Locate the flat field images and inspect them with ds9. Describe how the pixel intensities vary over the two-dimensional image. Are they flat? Compute the standard deviations of the pixel values in the images. There are two types of flats. Discuss why the flats are different. Use the bias (and dark frames) from Ex. 2.1 to make the basic corrections of the flat field images. Inspect the frames and describe what do you see. Generate a master flat field. Inspect it and verify that no stars are visible in the sky flats. Explain why you need to normalize this flat field image. How is this best done?

3. Verify and test that you have robustly normalized the master flat field: The best way to 'verifying' is to take a single raw flat field, and use the master-flat to 'flatfield your flat field'.

4. Quantify the variance of your master-flat: find a region that is essentially flat and measure the sample mean and standard deviation of the intensity. Consider how large a region to use. Repeat and make a similar investigation on a single flat frame. Compare the variances in the two flats

you've just measured. What have you learned? Justify that you have a robust determination of the pixel-to-pixel variations in this master flat.

5. Write a script to determine the gain and the RON using the equations on page 73 of Howell. Fr the flats you need to select two files using the same filter. It is critical to examine the files and find a good region of the detector for this calculation. Why can we determine the gain this way? Compare the output to the image header value. Re-check your results in Questions 4 and 9.

*Hint*: As explained in section E) in the introduction notes, and to solve the task of deriving the gain value of the detector you must convert your data to floats. One way to do this is to simply multiply your array data with 1.0 . E.g. array = array*1.0

Perhaps a better way to avoid this problem consistently, is to always read in the FITS files with data types as floats, e.g.:

```
>>>image = fits.getdata(filename , dtype='f')
```

## *Exercise 2.3: Processing of Science Images*

1. Pick science images in one filter (one filter for each of your group members, after which you can share the final processed FITS files). You just need to process these following science images:

```
EFOSC.2000-12-30T07:58:50.968.fits
EFOSC.2000-12-30T07:59:36.806.fits
EFOSC.2000-12-30T08:00:22.811.fits
```

as well as the associated photometric standard-star images. These files are found in the CALIB directory, and have OBJECT FITS header keywords identifying the name of the photometric field : PG 1323-086:

```
EFOSC.2001-01-03T08:54:27.270.fits
EFOSC.2001-01-03T08:55:18.187.fits
EFOSC.2001-01-03T08:56:03.899.fits
```

Correct for bias (and dark), and flat field these data and write the output to a new, procced FITS file. You can do this easily with simple numerical computations in a script, but you can also use the function `ccdproc.ccd_process`

When using the function, what is the output data units of the results?

The latter function computes the associated error per pixel in addition to the processed 2D array If you include a parameter "`error = True`" in the call of `ccd_process`. To work properly, then you also need to supply the gain and the RON value. Then your output will be in what units? This noise extension to the FITS image will be a useful check in the next exercise. To see how to display the FITS extension of the error, check Exercise 1 notes.

Check your results are reasonable: check the noise statistics, check that artifacts such as dust specs are corrected, etc. Verify that the resulting image has a uniform background. Evaluate which type of flat field is the best and justify your choice of flat field type.

### *Exercise 2.4: Error Propagation*

1. Write a script to propagate the errors and uncertainties of the data processing for all your relevant science frame (M67). Follow the instructions in section H. Start by converting all the relevant frames you need to work with into units of electrons – and keep all your calculations in electrons. But be careful: since the normalized flat field, *f,* is unitless it should not be converted to electrons. But the value $\bar{F}$ should be converted to electrons.

2. Evaluate and describe the error (and variance) image you get. Look at the various noise-contributions from source counts, read-noise, bias-correction, and flatfielding for the different band-passes. Which is largest? Is it the same for all band-passes?

3. Include in your report the `proof' of the scaling relevant for the flat field correction: Pick a region in your processed science image that contains just the sky background. Determine the r.m.s. scatter of the sky background in this region. Compare with the values you have in the exact same region in your noise image. Be aware of using the same units. Compare pixel-by-pixel and average values.

4. Discuss (also in the report) the optimal count level in each flat field exposure, which minimizes the noise of the final processing step. Contemplate and explain the difference between read-noise limited and sky-background limited observations.

5. Compare your computed noise (or variance) image with the noise image provided by the `ccdproc.ccd_process` function in question 2.3.1. Are the levels the same? Explain any differences in the two images.

6. Trim the reduced images and error images to smaller frames by trimming off both the right most and the leftmost 10 columns and the upper-most 20 rows of the 2d array. The function `ccdproc.trim_image` is convenient for this task. You should now have a final 2D FITS image with 1010 x 1010 pixels. Verify the result by displaying the images.

### *Exercise 2.5: Removing Cosmic Ray Hits*

This is an optional exercise:

Removing the cosmic ray hits in the science images is a step we will *not do* for the data used for exercises 1 – 4 because the 5 sec integration is so short that the number of cosmic ray hits are negligible. To test the functionality, you can obtain a reduced file with a long exposure time on Absalon : Look for the filename B_long.fits

1. The `ccdproc` package contains a function called `cosmicray_lacosmic` which comes from the LA Cosmic procedure described in van Dokkum (2001). Apply the task to the image and verify which cosmic rays are rejected, and that the stars are not rejected by changing the sigma-clipping level. Again, be aware that the output file has 2 FITS extension.

   For more information on the code see http://www.astro.yale.edu/dokkum/lacosmic/

Alternatively, when you have multiple frames obtained for a single field with a substantial spatial overlap between the images you have to combine the images to a single final frame.

- For each filter, determine image shifts from each frame to a reference frame. There are different ways of finding shifts: e.g. determining the center positions in x, and y of (several) stars common in the frames, or cross-correlations between images to find the best fit x,y offsets.

- Then combine all reduced images using the computed offset between each image. Verify that cosmic ray hits were rejected in the image combination, and that the algorithm does not reject the real objects (stars, galaxies) in your images.