Req 5.3.1

Title:

CCD dark current.

Objective:

Measure CCD dark current (in ADU/pixel/hour) for qualification purposes of the detector chain (qualification and trend analysis). The particle event rate will be determined on the fly.
Repeating the test with the dome lights on will provide information on possible light leaks.

Three one hour exposures are taken with the shutter closed. After rejection of the cosmic ray events, the signal above the bias level is the dark signal.
For the reduction of the science observations the subtraction of the sky brightness will include the dark current, and a separation of both contributions is normally not required.
Do a trend analysis.

Fulfilling or fulfilled by:

Selfstanding, also used to compute the particle event rate.

When performed/frequency:

Daytime (if dome and camera are proven to be light tight enough) - Commissioning; once per week. Alternatively, one dark frame per day could be taken, followed by a trend analysis once/month.

Sources, observations, instrument configurations:

Three 1 hour exposures with shutter closed. Dome lights either on or off for all exposures.

Inputs:

3 raw dark frames
CalFile– 541 *Master Bias frame*
Lab values
CalFile– 531 *Dark count rate for each CCD* older versions.
CalFile– 532 *Particle event rate* older version.

Outputs:

CalFile– 531 *Dark count rate for each CCD* in ADU/pixel/hour
CalFile– 532 *Particle event rate* in particles/cm^2/hour
In ESO DFS terminology these outputs are not a **CalFile**, but a single number.

Required accuracy, constraints:

Dark count rate should be less than the equivalent of 3 e^-/pixel/hour in ADUs excluding bad pixels. (This corresponds to 1.5 ADU/pixel/hour for nominal gain.)
Accuracy of determining particle event rate 1 particle/cm^2/hour.
Particle event rates should be identical for each chip.

Estimated time needed:

Observation: 3 hours. Reduction: $<$ 1 min./CCD.

Priority:

very important

TSF:

Mode– Stare N=3
TSF– OCAM_img_cal_dark exposure time 1 hour each, shutter closed. Dome lamp either on or off for all exposures

Recipe:

```
Dark_Current -i dark1 dark2 dark3 -b bias [-oc OVERSCAN_CORRECTION]

dark1 dark2 dark3   : 3 raw dark frames
bias                : master bias frame
OVERSCAN_CORRECTION : overscan correction mode (integer).
                      Description of allowed values:
                        0: apply no overscan correction (default)
                        1: use median of the prescan in the
                           x-direction
                        2: use median of the overscan in the
                           x-direction
                        3: use median of the prescan in the
                           y-direction
                        4: use median of the overscan in the
                           y-direction
                        5: use the per-row value of the prescan
in
                           the x-direction
                        6: use the per-row value of the overscan
in
                           the x-direction
```

Before applying this recipe, use **Recipe– Split**—which is documented in **seq.–631**—with the `-t dark` option to split the raw multi-extension FITS input files.

Needed functionality:

image - processing (eclipse.trim_and_overscan())
image - arithmetic (eclipse.image_sub_local())
image - statistics (eclipse.iter_stat())
sextractor - cosmic

CA:

Process (make):
1. Trim, de-bias and overscan correct the input images
2. Median average the 3 debiased images.
3. Iteratively reject 5 σ outliers in the median image.
4. Compute the mean of the resulting image.
5. Use this mean to compute the dark current in ADU/pixel/hour.
6. For each de-biased and overscan corrected input image:
6.1. Use sextractor with a cosmic retina filter to create a catalog.
6.2. Count the number of objects in the catalog.
7. Use the event counts to compute the particle event rate in particles/cm^2/hour.

Verification (verify):
1. The dark current should be less than 2 ADU/pixel/hour.
2. The difference in event counts between the 3 measurements should be less than 3 σ

Trend Analysis (compare):
1. The difference between consecutive dark current measurements should be less than 1.0 ADU/pixel/hour.

CAP:

```
Constants:
  MAXIMUM_ITERATIONS              : statistics measurement (in-
teger)
                                    Value: 3
  REJECTION_THRESHOLD             : sigma rejection for bad pix-
els (float)
                                    Value: 5.0
```

```
  PIXELSIZE                        : OmegaCAM pixelsize in cm (float)
                                     Value: 15e-4
  COSMIC_CONFIG : SExtractor configuration parameters and their
values
                  in case they differ from the SExtractor defaults.
          CHECKIMAGE_TYPE = 'SEGMENTATION'
          ANALYSIS_THRESH = 3.0
          CATALOG_TYPE    = 'ASCII'
          DETECT_MINAREA  = 1
          DETECT_THRESH   = 6.0
          GAIN            = 1
          MEMORY_PIXSTACK = 100000
          BACKPHOTO_THICK = 24
          BACK_TYPE       = 'AUTO'
          BACK_VALUE      = 0.0, 0.0
          FILTER          = 'Y'
          FILTER_NAME     = 'cosmic.ret'    # Retina filter for
SExtractor
          CHECKIMAGE_NAME = 'cosmic.dat'   # Image with cosmic ray
events


# preprocessing
processed = []
darks = [dark1, dark2, dark3]
for dark in darks:
    ima = eclipse.trim_and_overscan(dark)
    eclipse.image_sub_local(ima, bias)
    processed.append(ima)

# compute dark current
median_image = eclipse.cube_average_median(processed)
stats = eclipse.iter_stat(median_image,
                          MAXIMUM_ITERATIONS,
                          REJECTION_THRESHOLD)
exptime = dark1.EXPTIME
dark_current = stats.avg_pix / exptime * 3600.0
```

```
# compute the cosmic ray counts
event_counts = []
particle_event_rate_sum = 0.0
for ima, dark in zip(processed, darks):
    sextractor(ima, COSMIC_CONFIG)
    count = line_count('cosmic.dat')
    event_counts.append(count)
    count /= dark.EXPTIME/3600.0
    count /= image.xsize()*PIXELSIZE*ima.ysize()*PIXELSIZE
    particle_event_rate_sum += count

# average the individual particle event rates
particle_event_rate = particle_event_rate_sum/3

# verification
if dark_current > MAXIMUM_DARK_CURRENT:
   EVENT_RATE_TOO_HIGH = 1
if (abs(event_counts[0]-event_counts[1]) >
    3 * sqrt(event_counts[0]+event_counts[1]):
   INCONSISTENT_EVENT_COUNTS = 1
if (abs(event_counts[0]-event_counts[2]) >
    3 * sqrt(event_counts[0]+event_counts[2]):
   INCONSISTENT_EVENT_COUNTS = 1
```