

Calibrations Overview

Francisco Paz Chinchón

Precal

Run nightly calibrations over domeflats and bias. Beware the code automatically discards first exposure of each filetype of the night, for safety. The nightly processing should run periodically, to make generation of supercal smoother.

Supercal

Assessments

1. Set of codes to assess domeflats using Wavelet in 2D
 - a. **Repository:** https://github.com/paztronomer/domeflat_calibrations
2. Time domain: the table `FLAT_QA` has some useful quantities to create time series and explore stability of nights
3. Time domain codes
 - a. **Repository:** https://github.com/paztronomer/sandbox_timedomain
 - b. **Repository:** https://github.com/paztronomer/structure_timedomain
4. Useful for detecting features when analyzing large sets of images:
 - a. DBSCAN
 - b. Auto-similarity (see codes in the above repositories)
 - c. use matplotlib for sequential display of a lot of footprint images
5. Remember to do visual inspection of individual domeflats. These are binned focal plane images (binned_fp filetype)
6. For specific features smaller than 128 pixels, is much better to use the whole CCD to get statistics

Note: Remember to delete precal runs after creating the Supercal, to save disk space.

Skytemplates

For a step-by-step, queries to select DB products, and discussion, refer to the below repository. Remember visual inspection of each full-sized CCD must be done, looking for structures.

Repository and documentation: https://github.com/paztronomer/skytemplates_build

Skytemplates generation for u-band

For the step-by-step, codes, and discussion, refer to the below repository. This was done retrieving additional DECam data from NOAO.

Repository and documentation: https://github.com/paztronomer/uband_endeavor

Some investigation for the wedge appearing in skytemplates

The notebook and additional codes can be found in the repository below.

Repository: https://github.com/paztronomer/skytemplates_wedge_mask

BPM

Use ~50 g-band exposures, following the step-by-step described in the following repository. Remember to do a bit-by-bit comparison against past epochs, to check for some out of the normal variation.

Repository and documentation: https://github.com/paztronomer/bpm_calibrations

Dilation of tapebumps masks

- It is a good guess to use same binary-dilation (numpy) number of pixels than the wide of the kernel used for assess profiles.
- Techniques I've used for detecting the mask regions to be enlarged:
 - median images at different levels of counts
 - segmentation using Otsu threshold

Add mask to BPM

Repository: https://github.com/paztronomer/bpm_addmask

Translate from a set of bit-definitions to another

Repository and documentation: https://github.com/paztronomer/bpm_translation

Check masks

Repository: https://github.com/paztronomer/mask_check

Calibrations for Y4 onwards

Visit: [Calibration products \(Y4 onwards\)](#)