Nebencal code update

Anne Bauer
February 28, 2014

Ubercal algorithm: relative calibration

AN IMPROVED PHOTOMETRIC CALIBRATION OF THE SLOAN DIGITAL SKY SURVEY IMAGING DATA

Nikhil Padmanabhan,[1,2,3] David J. Schlegel,[1,4] Douglas P. Finkbeiner,[4,5] J. C. Barentine,[6,7] Michael R. Blanton,[8] Howard J. Brewington,[7] James E. Gunn,[4] Michael Harvanek,[7] David W. Hogg,[8] Željko Ivezić,[9] David Johnston,[4,10] Stephen M. Kent,[11] S. J. Kleinman,[7,12] Gillian R. Knapp,[4] Jurek Krzesinski,[7,13] Dan Long,[7] Eric H. Neilsen, Jr.,[7,11] Atsuko Nitta,[7,14] Craig Loomis,[4,7] Robert H. Lupton,[4] Sam Roweis,[15] Stephanie A. Snedden,[7] Michael A. Strauss,[4] and Douglas L. Tucker[11]

Received 2007 March 20; accepted 2007 October 8

$$m = m_{\rm ADU} + a_\alpha - \left[k_\beta + \left(\frac{dk}{dt}\right)_\beta (t - t_{\beta,\rm ref})\right] x + f_\gamma(j), \quad (6)$$

$$\chi_i^2 = \sum_{j \in \mathcal{O}(i)} \left[\frac{m_i - m_{j,\rm ADU} - a_{\alpha(j)} + k_{\beta(j)}(t)x - f_{\gamma(j)}}{\sigma_j}\right]^2, \quad (8)$$

$$\chi^2 = (\boldsymbol{A}\boldsymbol{p} - \boldsymbol{b})^t \boldsymbol{C}^{-1} (\boldsymbol{A}\boldsymbol{p} - \boldsymbol{b}), \quad (12)$$

$$\frac{d\chi^2}{d\boldsymbol{p}} = \boldsymbol{A}^t \boldsymbol{C}^{-1} \boldsymbol{A}\boldsymbol{p} - \boldsymbol{A}^t \boldsymbol{C}^{-1} \boldsymbol{b} = 0. \quad (14)$$

zeropoints

dependence on position, airmass, etc.

mag errors

mag measurements

matrix equation: Ax = b

Algorithm

- Calibrate assuming that the weighted mean of “good” mags is the truth
    - Minimize relative offset between observations
- Absolute calibration
    - Include PreCam (and/or SDSS, ...) standards as if from one image
    - At the end, normalize all ZPs -= ZP(PreCam) to make a calibration pegged to the PreCam data

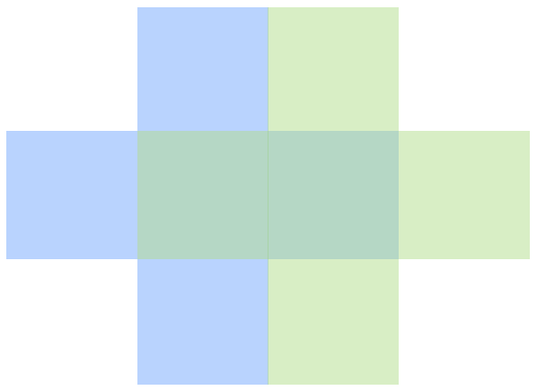
Nebencal algorithm

- Inspired by LSST's implementation
- Limit memory usage by pixelizing the sky via Healpix
- Calibrate each pixel plus neighbors in one übercal (~tens of degrees, customizable size)

- At the end, do another übercal to normalize these pixel regions together

Nebencal algorithm

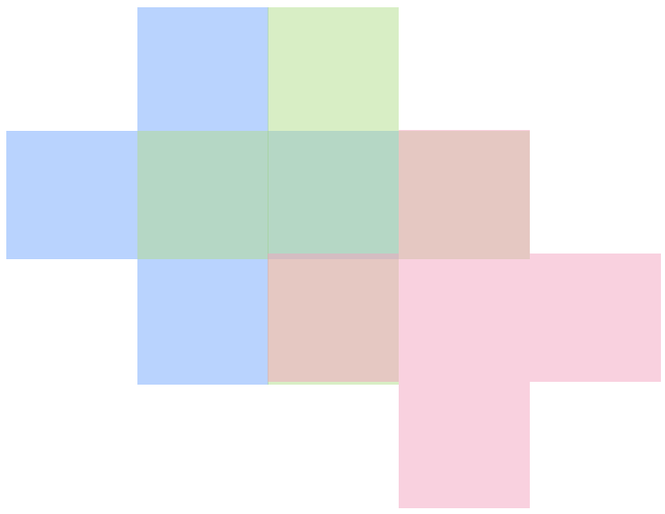
- Inspired by LSST's implementation
- Limit memory usage by pixelizing the sky via Healpix
- Calibrate each pixel plus neighbors in one übercal (~tens of degrees, customizable size)



- At the end, do another übercal to normalize these pixel regions together

Nebencal algorithm

- Inspired by LSST's implementation
- Limit memory usage by pixelizing the sky via Healpix
- Calibrate each pixel plus neighbors in one übercal (~tens of degrees, customizable size)



- At the end, do another übercal to normalize these pixel regions together

Nebencal algorithm

- Inspired by LSST's implementation
- Limit memory usage by pixelizing the sky via Healpix
- Calibrate each pixel plus neighbors in one übercal (~tens of degrees, customizable size)

- At the end, do another übercal to normalize these pixel regions together

Implementation: Config yaml file

```yaml
general:

        # what filter(s) do we want to calibrate?
        filter : 'z'

        # note that these input files depend on the filter!
        precam_filename : "/Users/bauer/surveys/DES/precam/PreCamStandarStars/Z.Stand1percent.s"
        zp_phot_filename : "/Users/bauer/surveys/DES/zp_phots/z.dat"

        # where are the inputs?
        globals_dir : '/Users/bauer/surveys/DES/y1p1/equatorial'
        sdss_filename : "/Users/bauer/surveys/DES/y1p1/equatorial/sdss/SDSSDR10_SouthGalCap/stripe82_sample1.csv"

        # include precam and/or sdss as standards?
        use_precam : True
        use_sdss : False

        # what nside is used in the global object file pixelization scheme?
        nside_file : 32


calibrations:

        - id_string : 'exposureid'
          outfilename : 'nebencal_exp_zps_z'
          nside : 1
          max_dets : 2500
          require_standards : True

        - id_string : 'image_id'
          outfilename : 'nebencal_img_zps_z'
          nside : 32
          max_dets : 500
          require_standards : False
```

Implementation: a couple of details

- First step, pre-calibration: match across exposures and make global objects. Saved in one binary file per Healpix pixel (in the highest pixel resolution I plan on using. The code degrades this file resolution to whatever is desired.)

    - Per detection: mag_psf & error, image & exposure ids, rasicam info. as yet unused: x, y, ccd, everything else in the DB...

- Next, just run: nebencal.py config.yaml

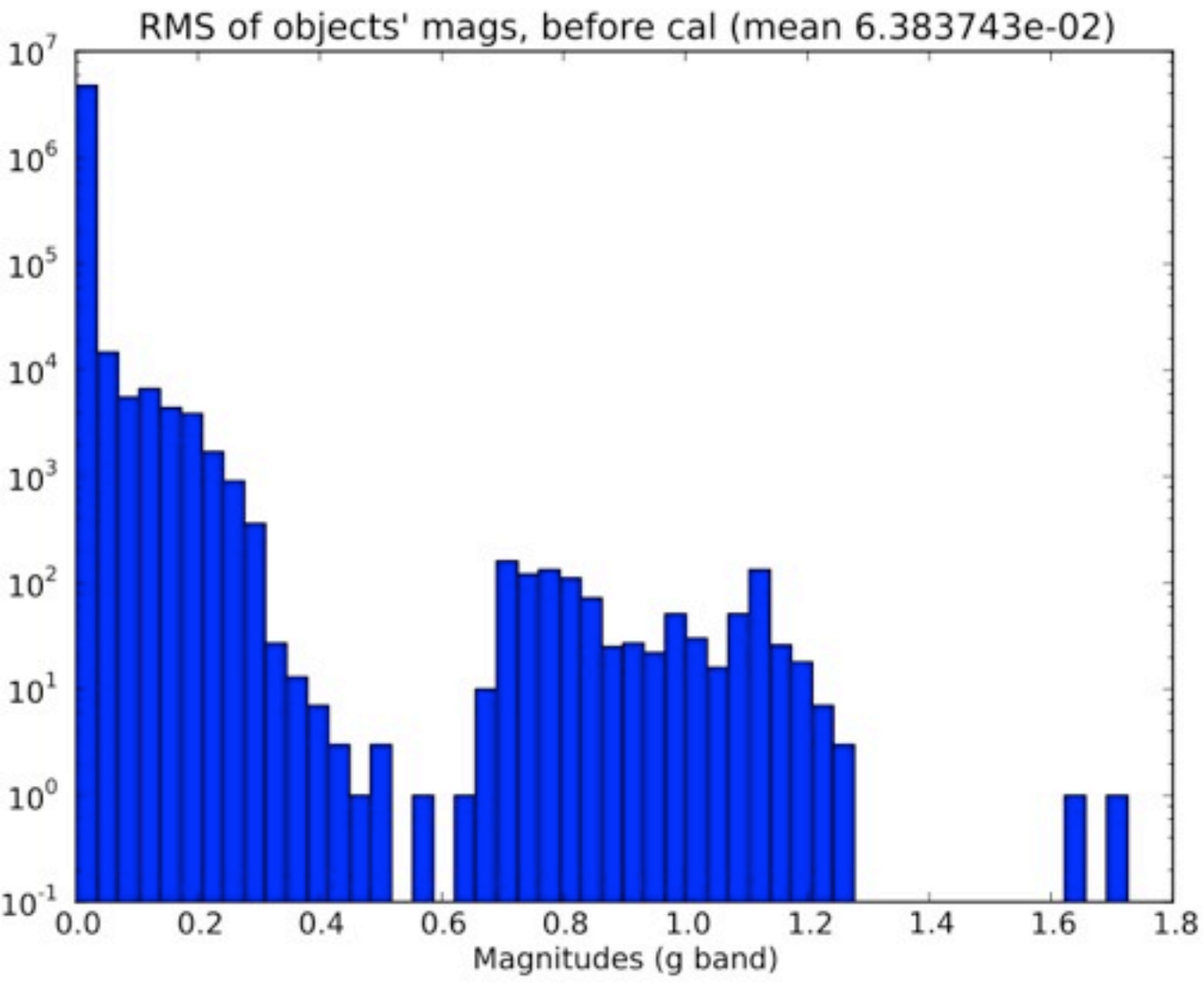
- I have my code in a git repository, let me know if you'd like access.

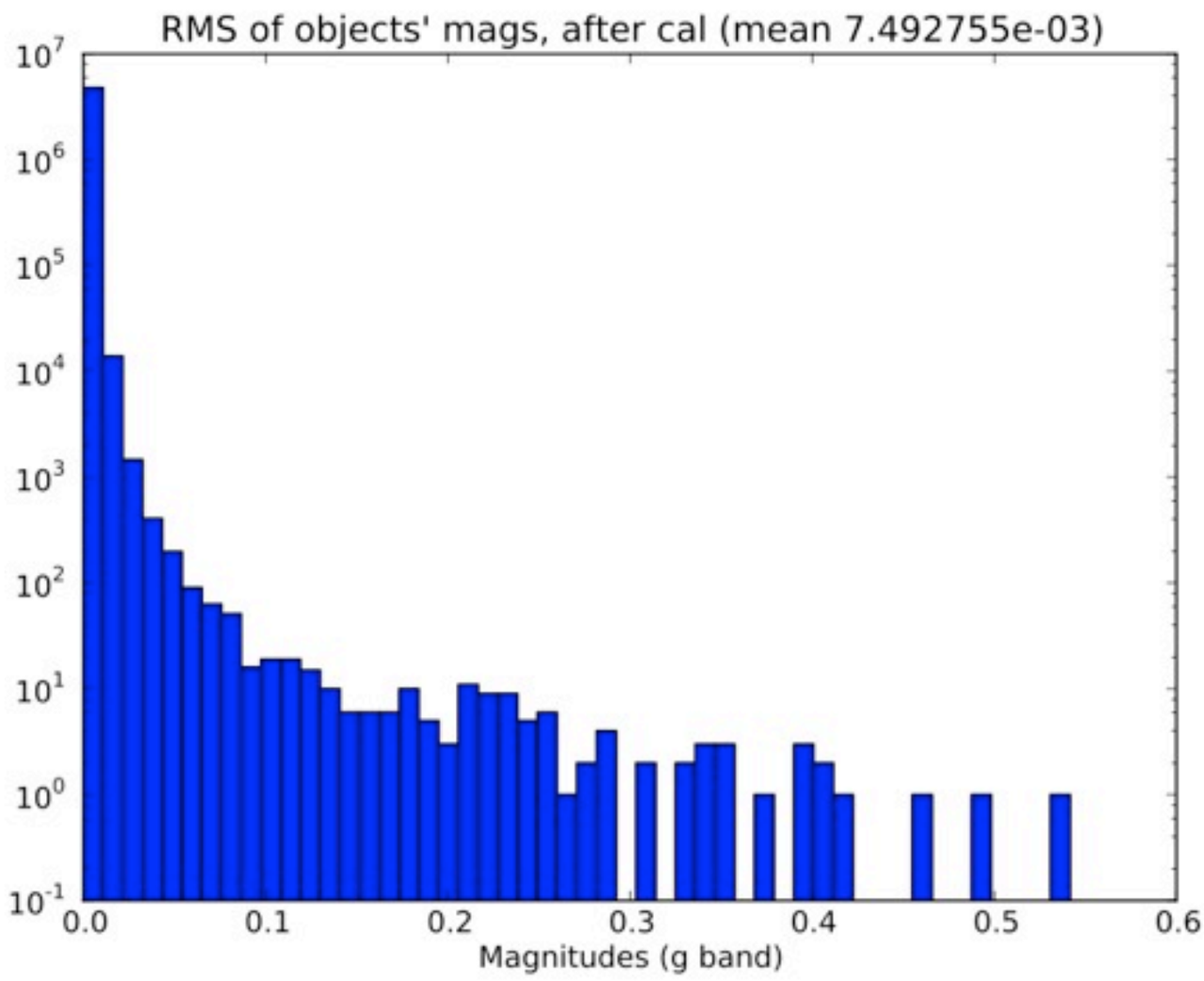
Implementation: a couple of details

- Written in python
- Uses sparse matrices to do the work: scipy.sparse.linalg
- The exposure calibration takes some (4-5ish) hours on one processor. The image calibration takes <1 hour. (For the equatorial region, per filter, without cuts on number of detections)
- Python multiprocessing module doesn’t work with CBLAS functions. I may try harder to parallelize the code.
- Takes ~6GB memory per run (exposure calibration limited), but I think it should be 2-3x less...

g band RMS of each objects' mag measurements

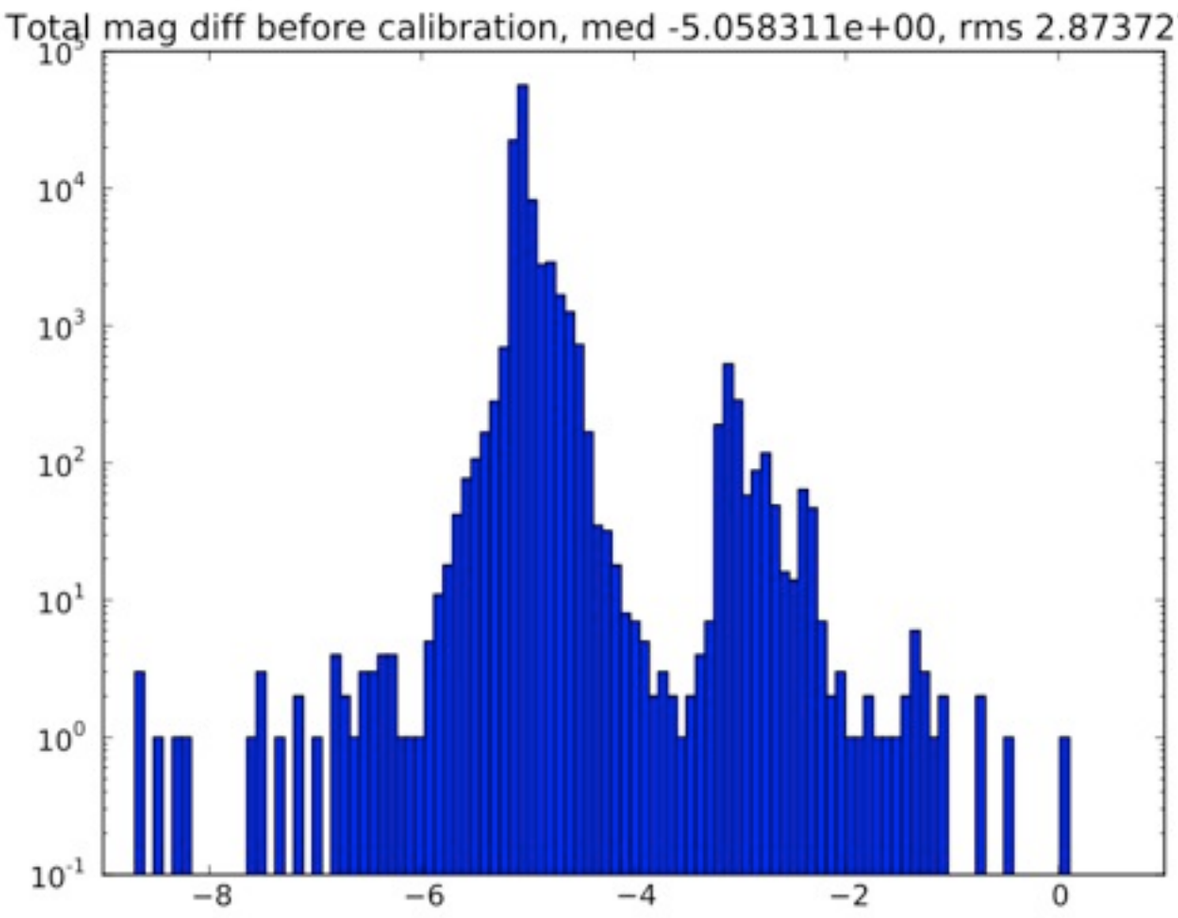
- Relative mag precision 6.4% calibrated to 0.75%

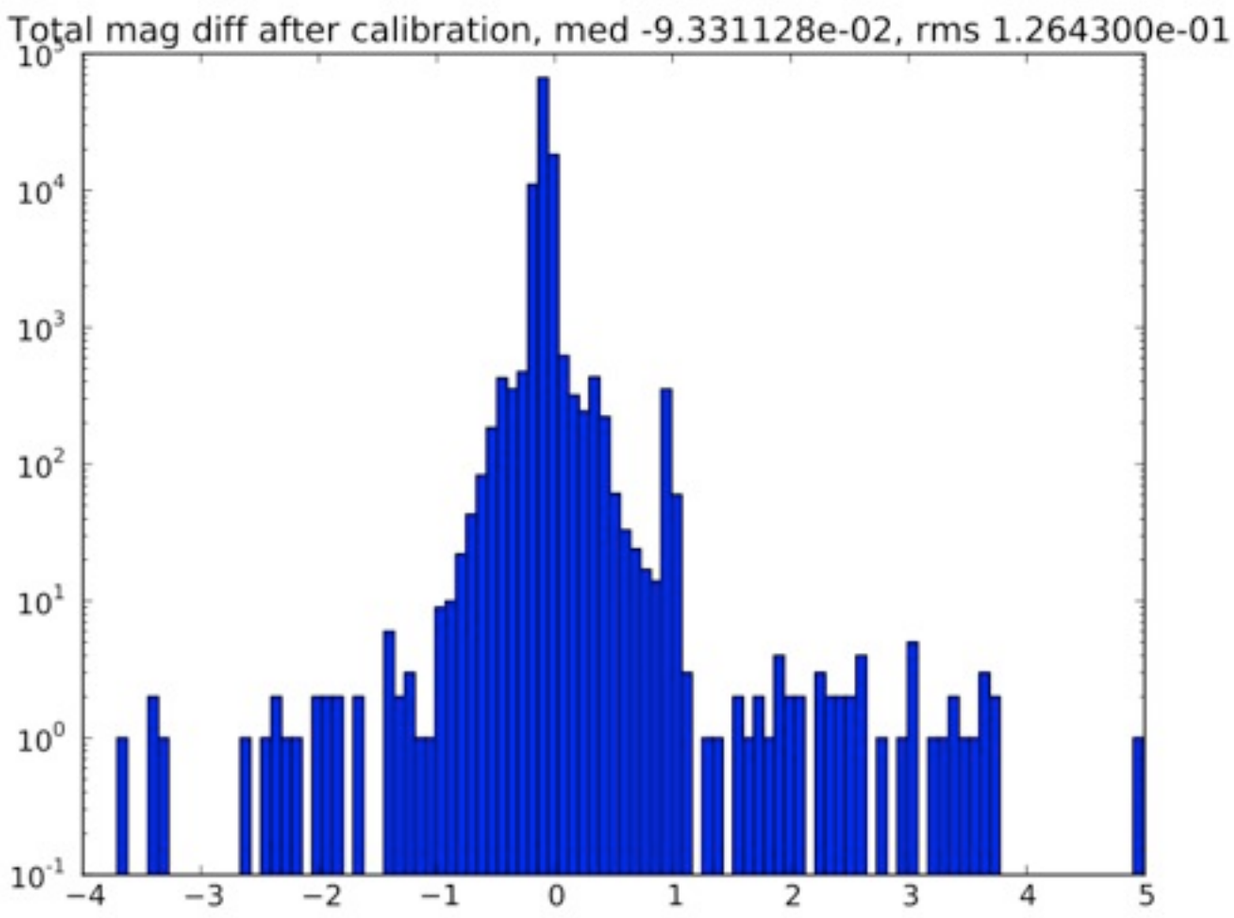




Absolute calibration is working (but sensitive to outliers)

- g-band using PreCam as standard objects, checking against SDSS as “truth”

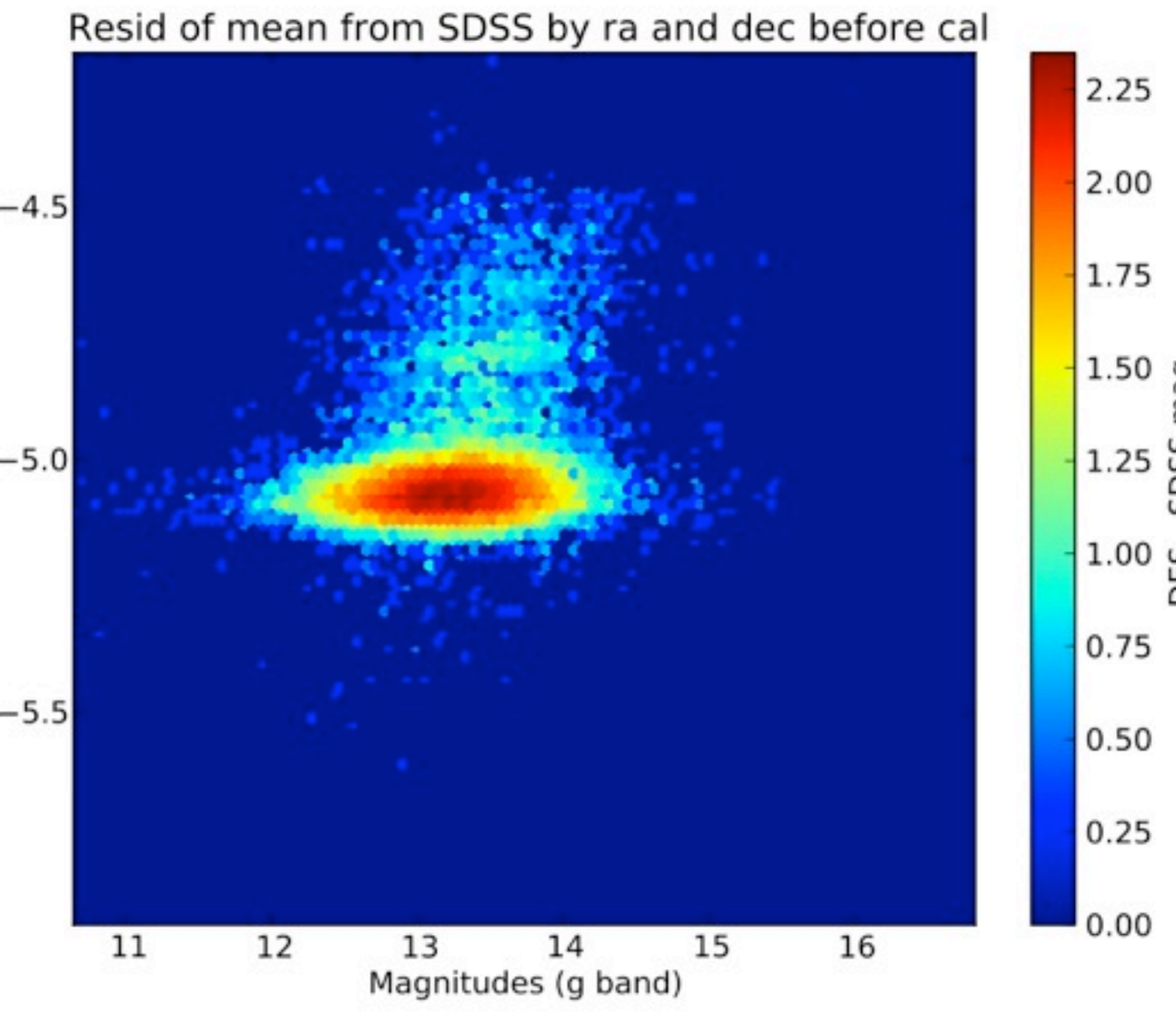


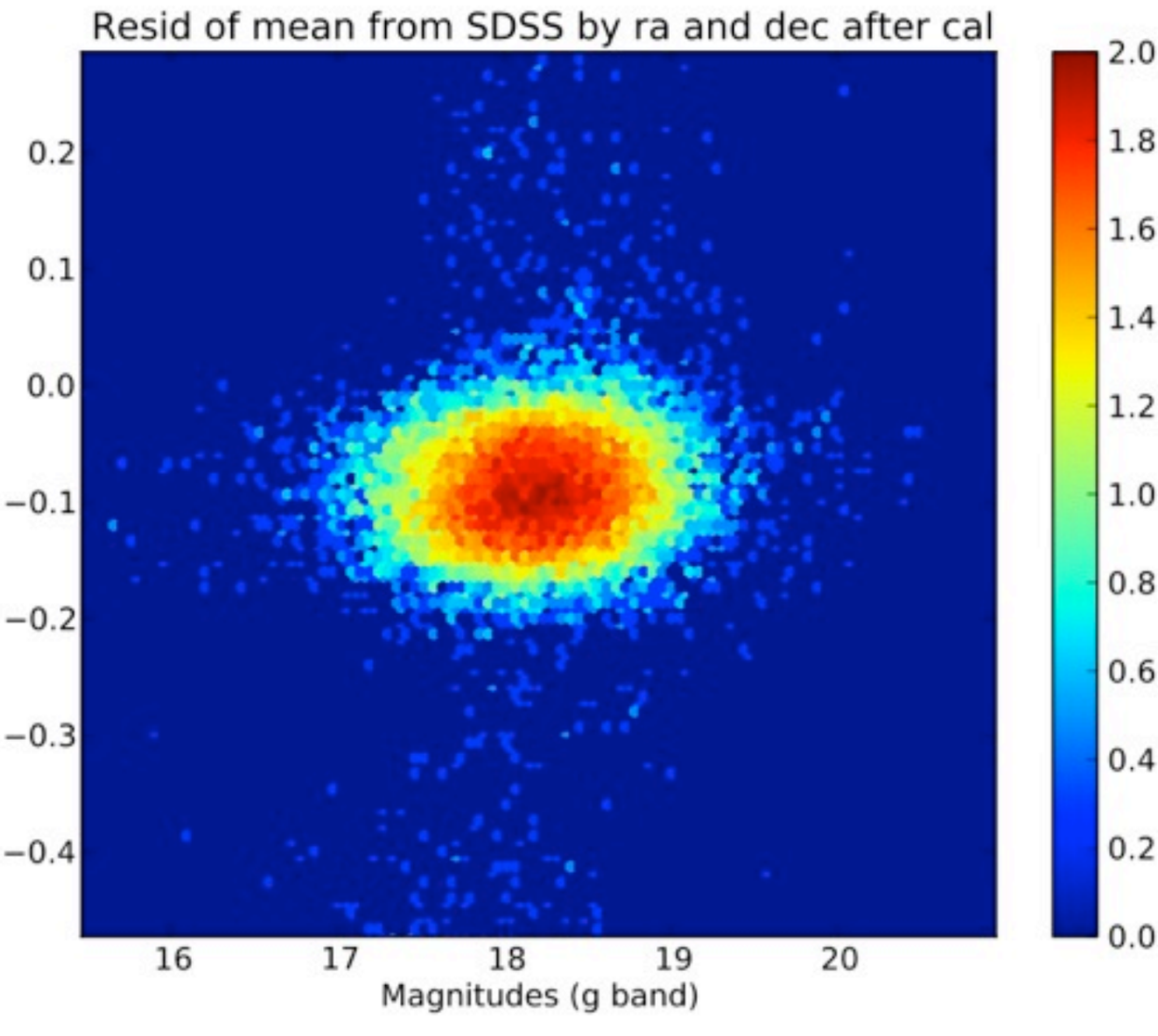


- Note final median is still off due to outliers... a solution is in the works.

Absolute calibration vs DES magnitude

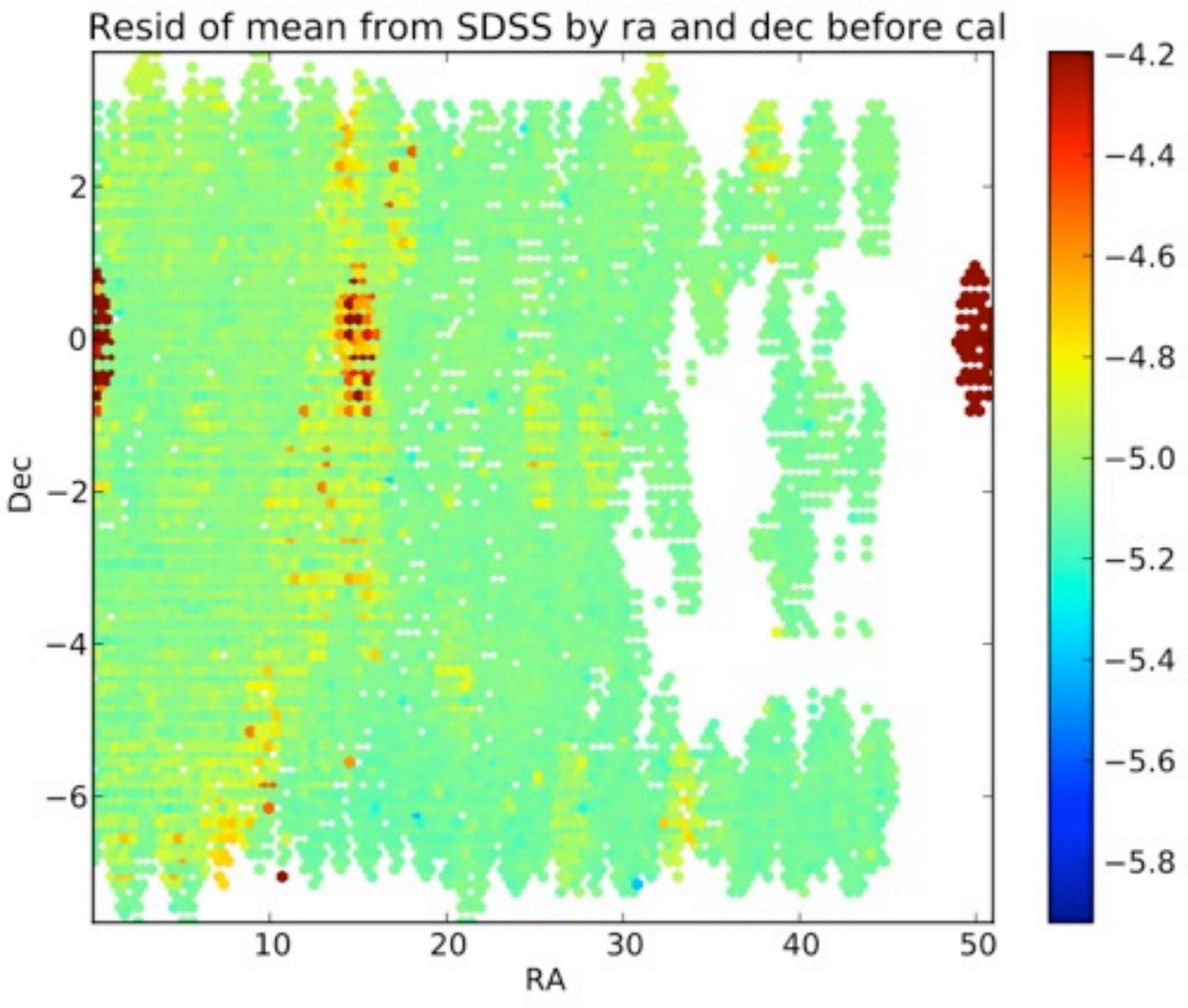
- DES-SDSS vs DES, before and after g-band calibration

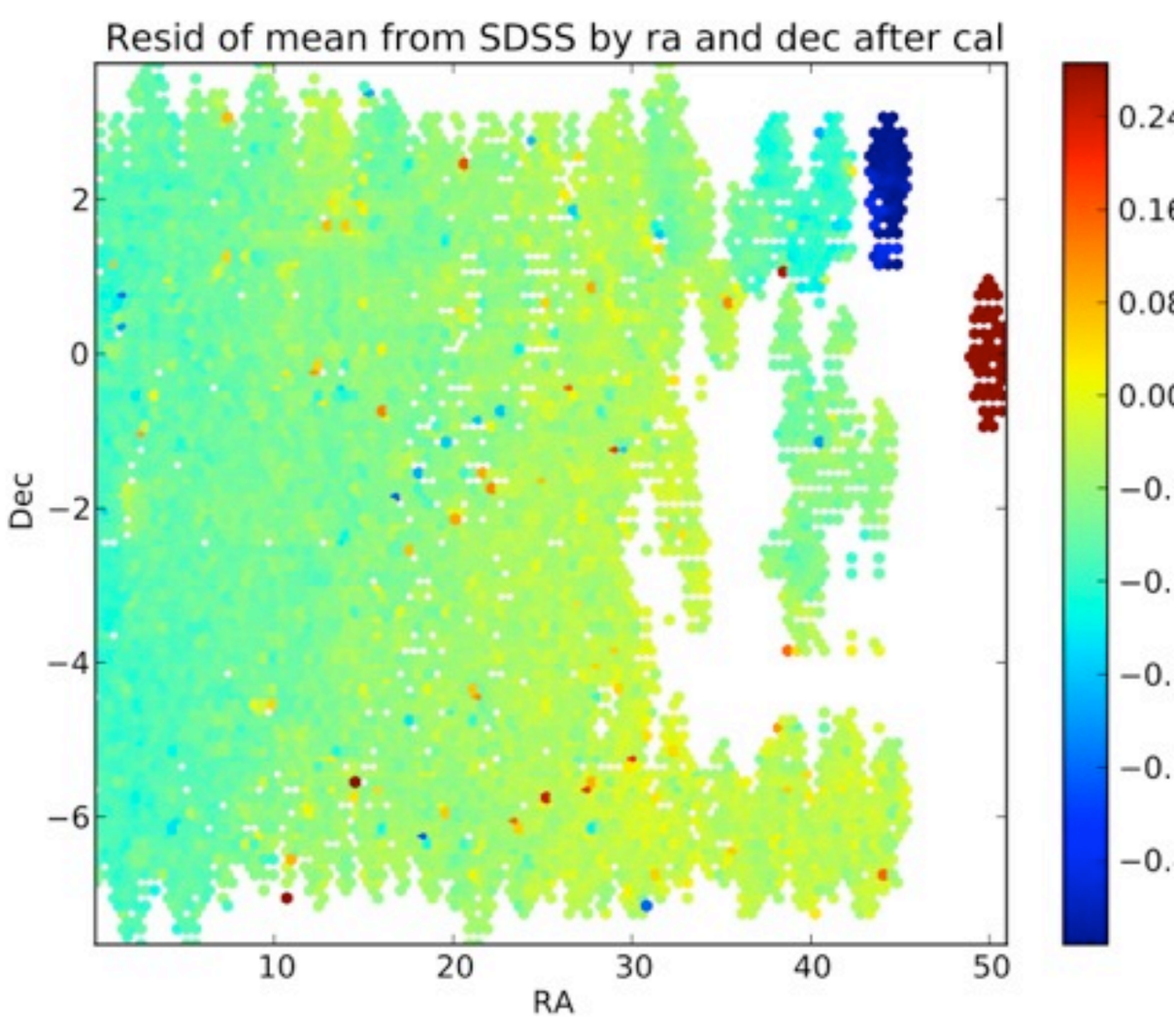




Position dependence of absolute residuals

- The worst residuals are from disjoint regions

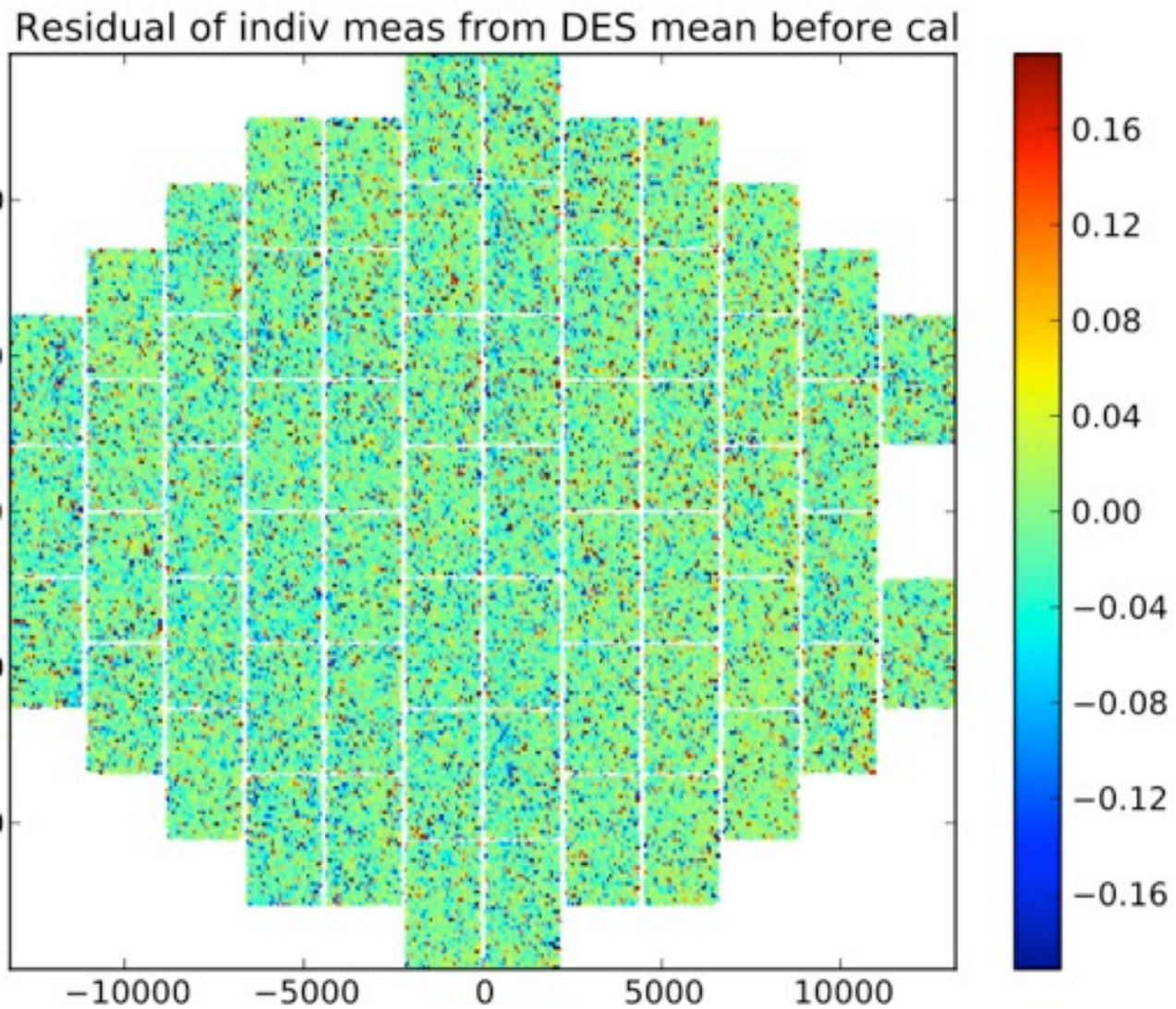


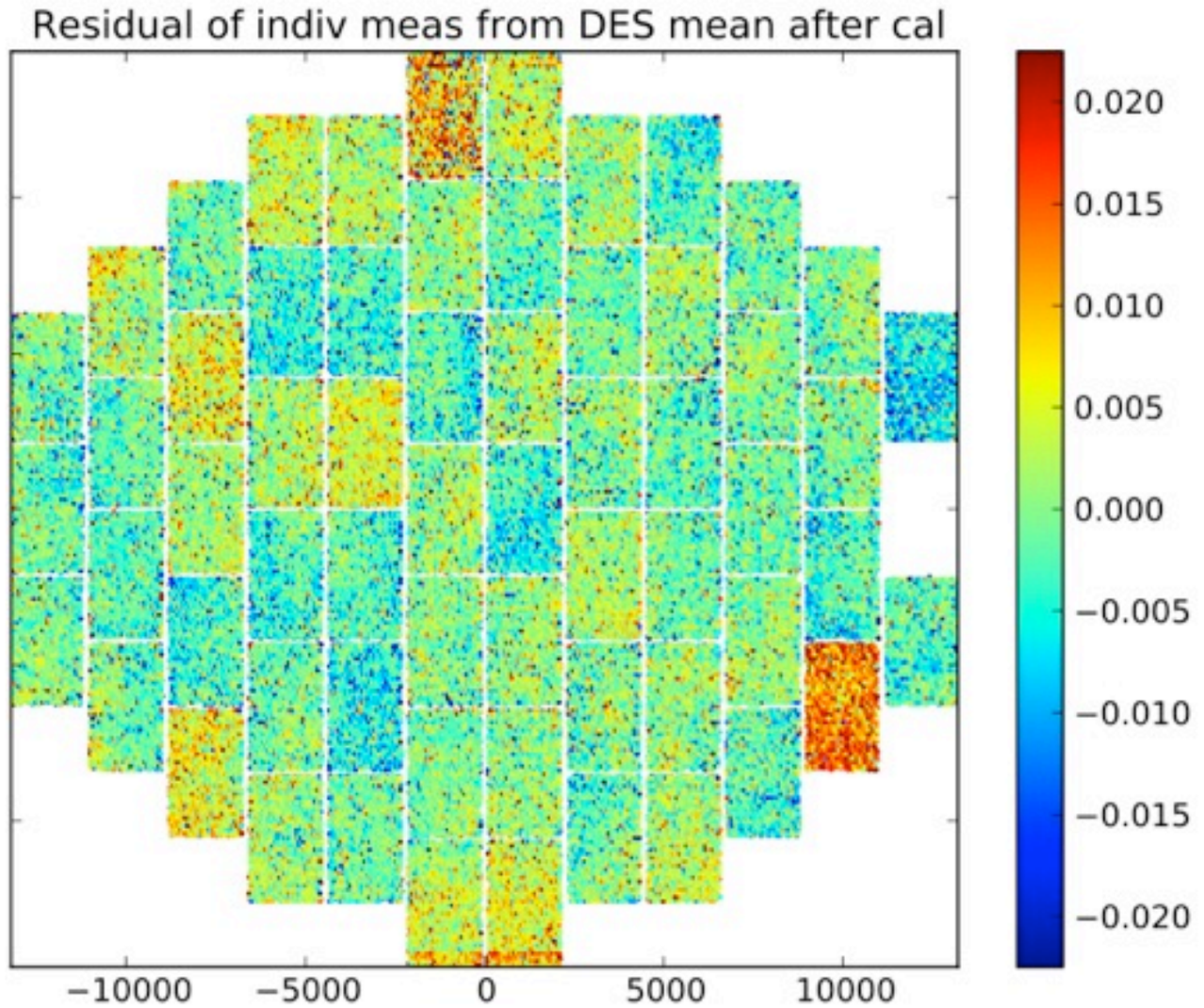


Plan to check for disjoint regions

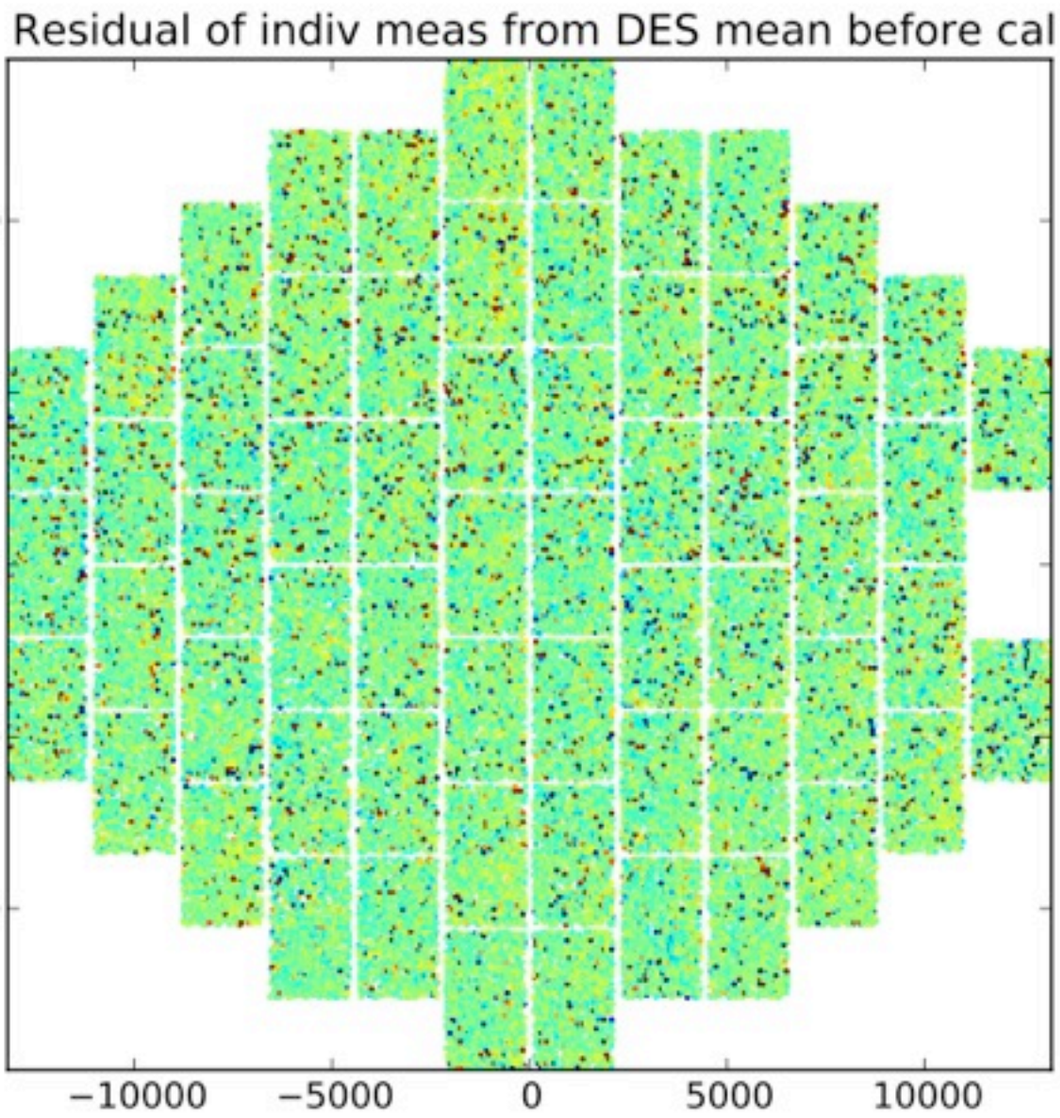
- I’m putting in a check to see if the graph of, e.g., image_id connections contains disconnected regions
  - `scipy.sparse.cs_graph_components`
- If we require coverage by standards and a region is disjoint from the standards, it will be removed from the calculation and flagged uncalibrated.
- Should help outlier rate and absolute calibration accuracy

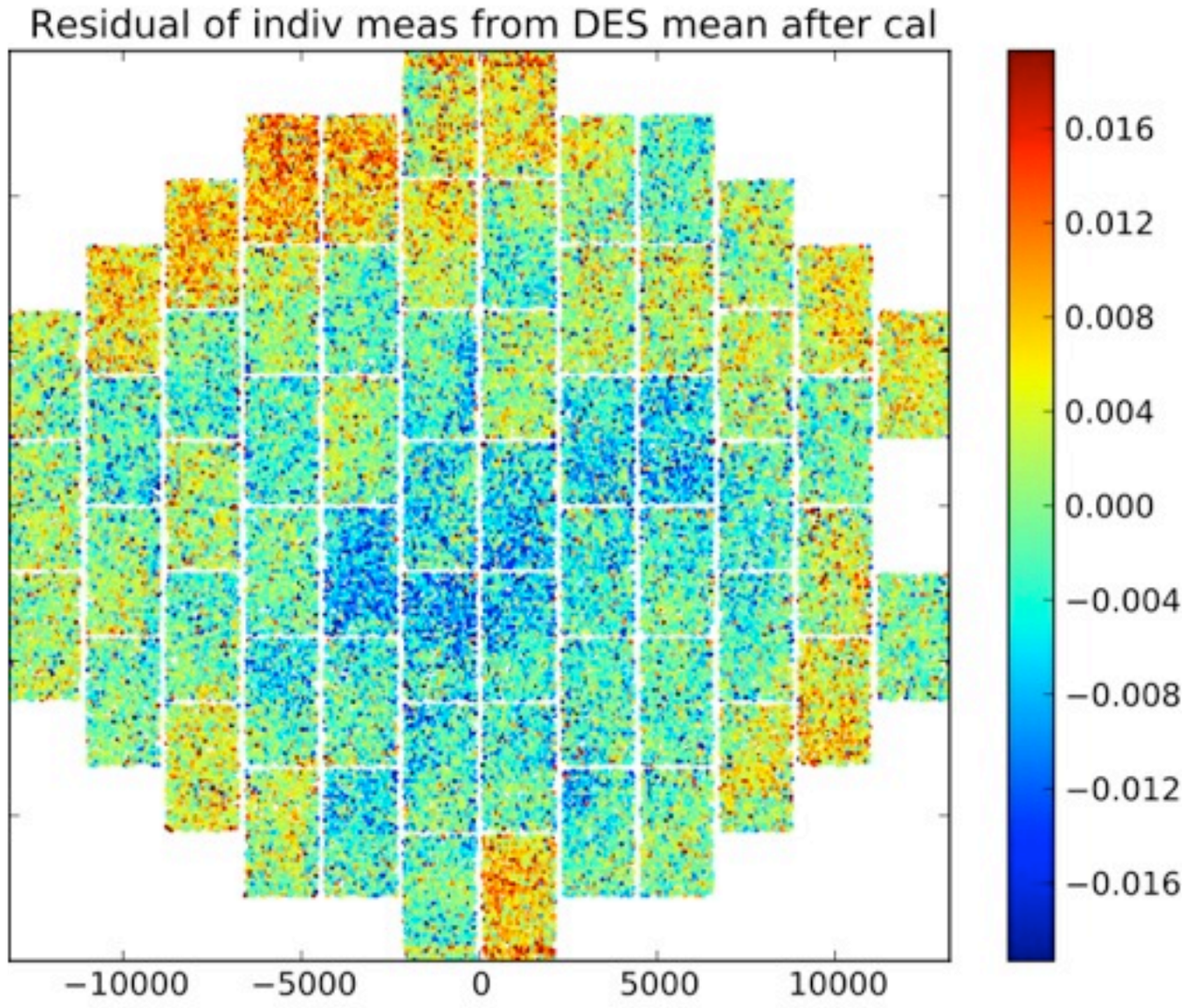
Focal plane dependence of residuals: g band



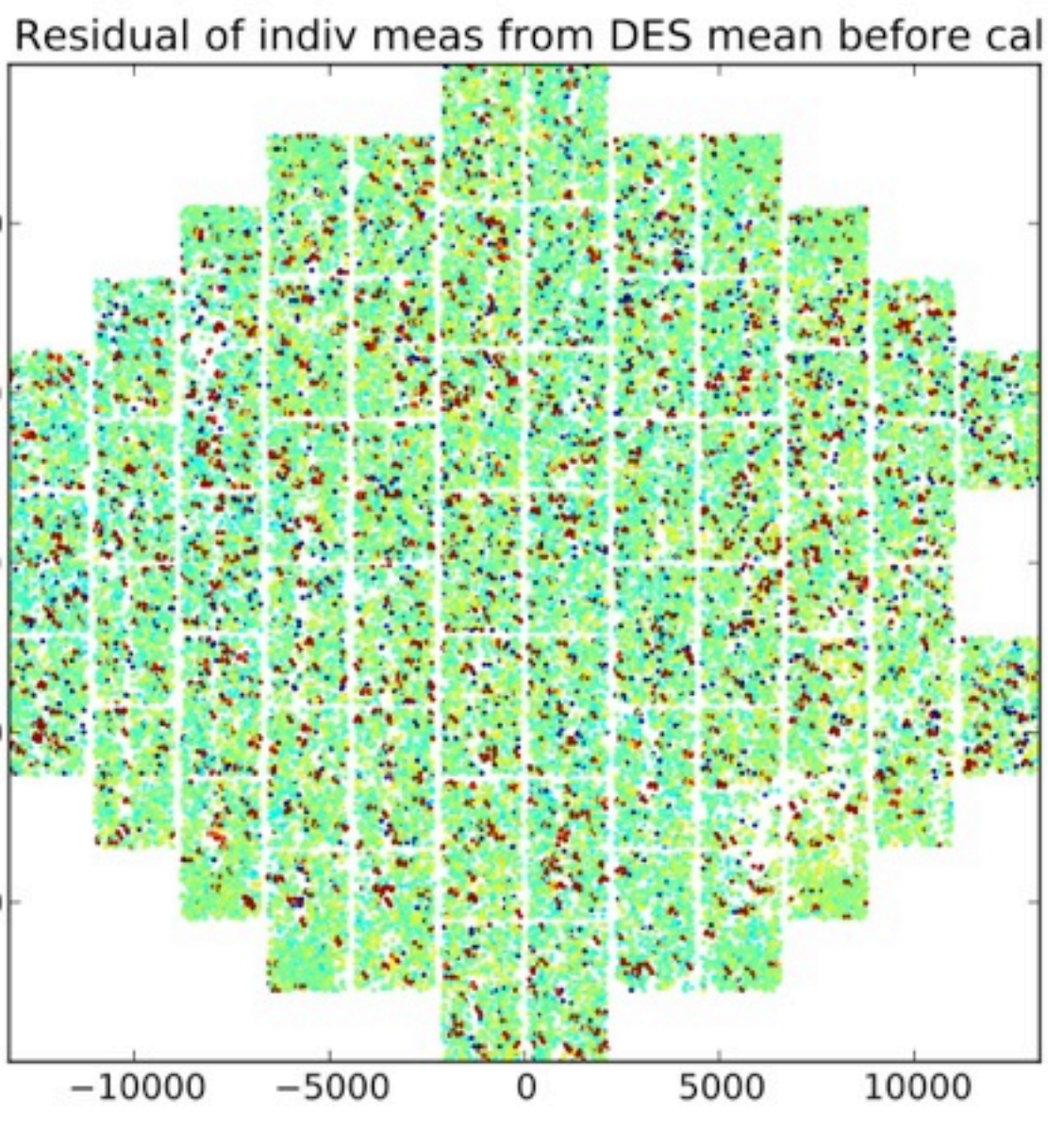


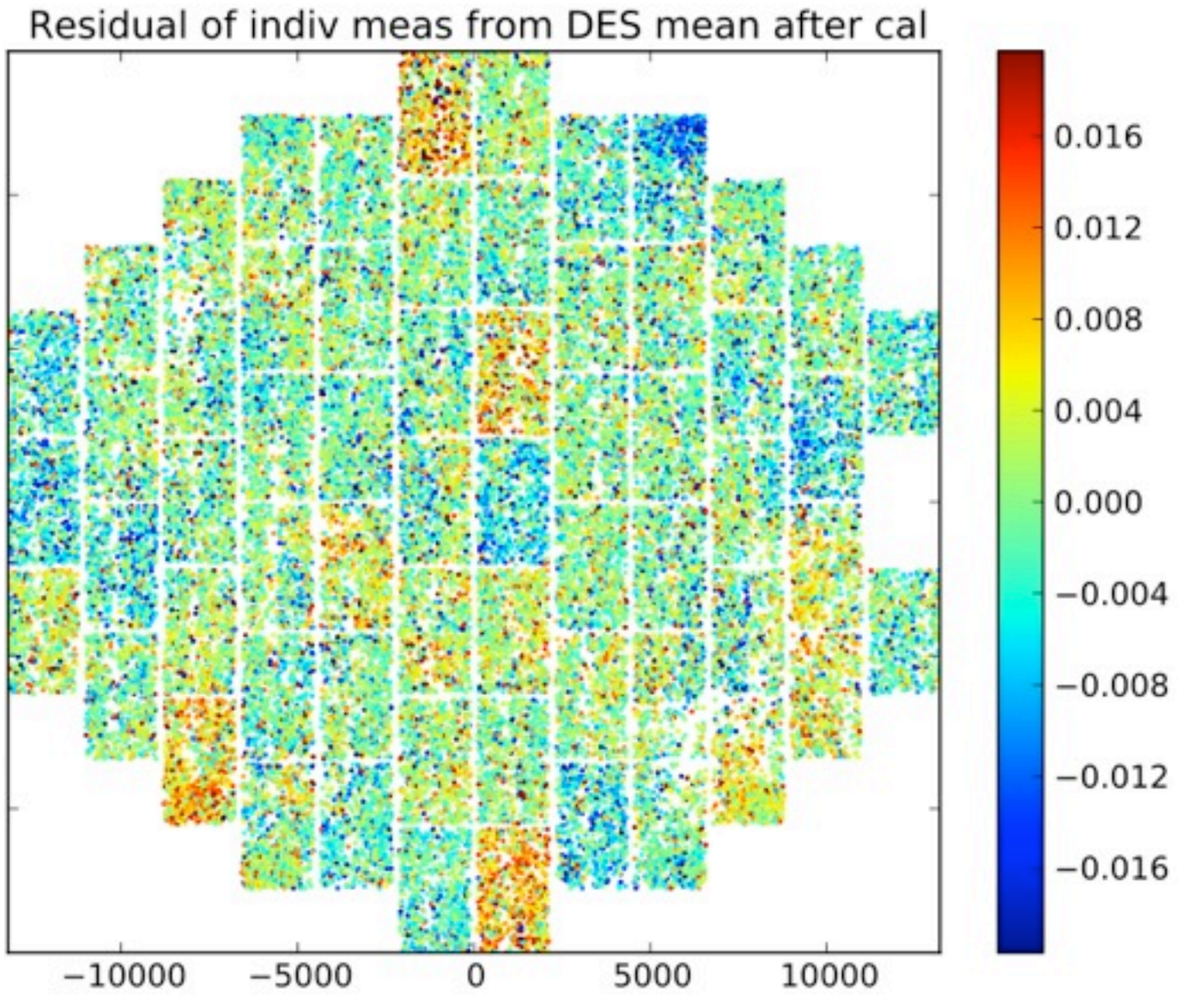
Focal plane dependence of residuals: i band





Focal plane dependence of residuals: g band





Future work

- outlier rejection on variable stars, bad measurements, etc.
- could compute zp error estimates
- exclude edges of chip? additional quality cuts?