Data reduction II
Photometry with IRAF

Harry Dawson

Research workshop on evolved stars
August 2024

Introduction

- Text in **yellow** is for typing into IRAF terminal
- In my slides IRAF terminals are black or papaya colour

Why data *reduction*?

- We need to subtract – or reduce – instrumental effects and background contamination.

Reducing instrumental effects:

- **BIAS**: image with ‘zero’ exposure time.
  (not possible with Perek! We will use the overscan region)
- **FLAT**: image of a uniformly illuminated surface.
  Estimate sensitivity difference throughout the CCD + dust grains, scratches etc.
- **DARK**: image with the same exposure time of the science image with the shutter closed.
  Estimate the level of background current.

More is more

- Each of the counts on the images has an associated uncertainty.
- If we take n images, each with an uncertainty σ_i, the uncertainty on the average will be $\sigma_i/\sqrt{n}$.
- Therefore, the first step in data reduction is to calculate the average for BIAS, FLAT, and DARK images.

More is more

- Each of the counts on the images has an associated uncertainty.
- If we take n images, each with an uncertainty σ_i, the uncertainty on the average will be $\sigma_i/\sqrt{n}$.
- Therefore, the first step in data reduction is to calculate the average for BIAS, FLAT, and DARK images.

  BIAS: not available for the Perek telescope.

  FLAT: master flat has already been created.

  DARK: we need to calculate the median dark.

IRAF – **I**mage **R**eduction and **A**nalysis **F**acility

- We'll use the package

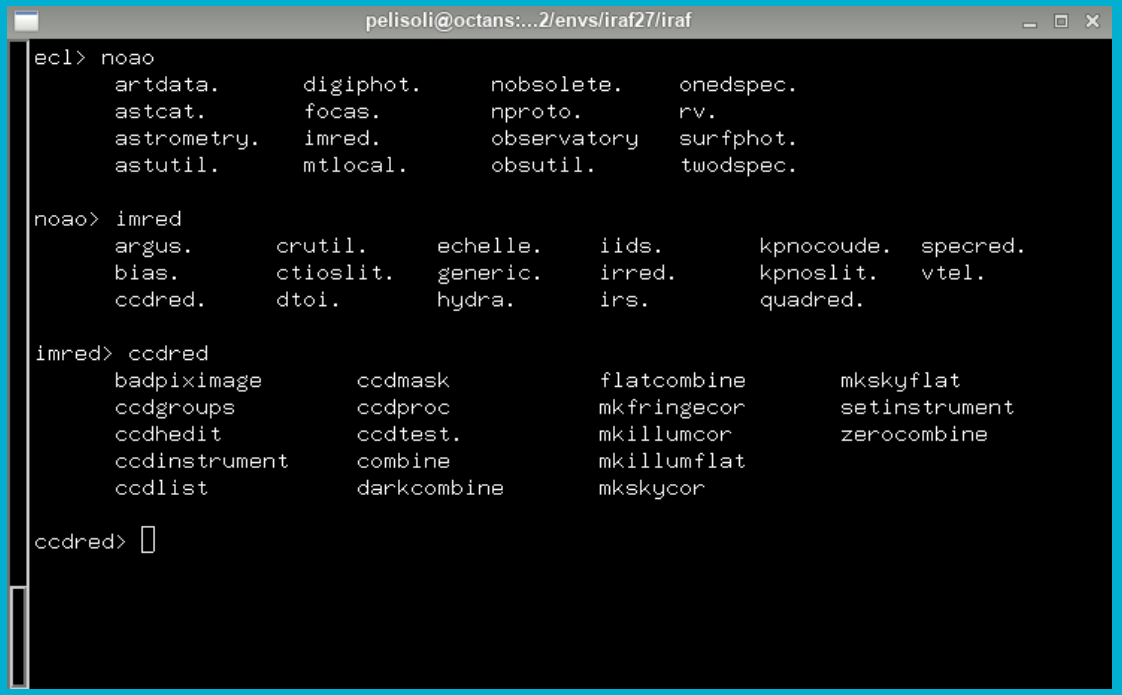
  noao imred ccdred

for the data reduction, and

noao digiphot daophot

for the photometry.

- Load each part of the packages by typing their name followed by enter.



Preparing working directory

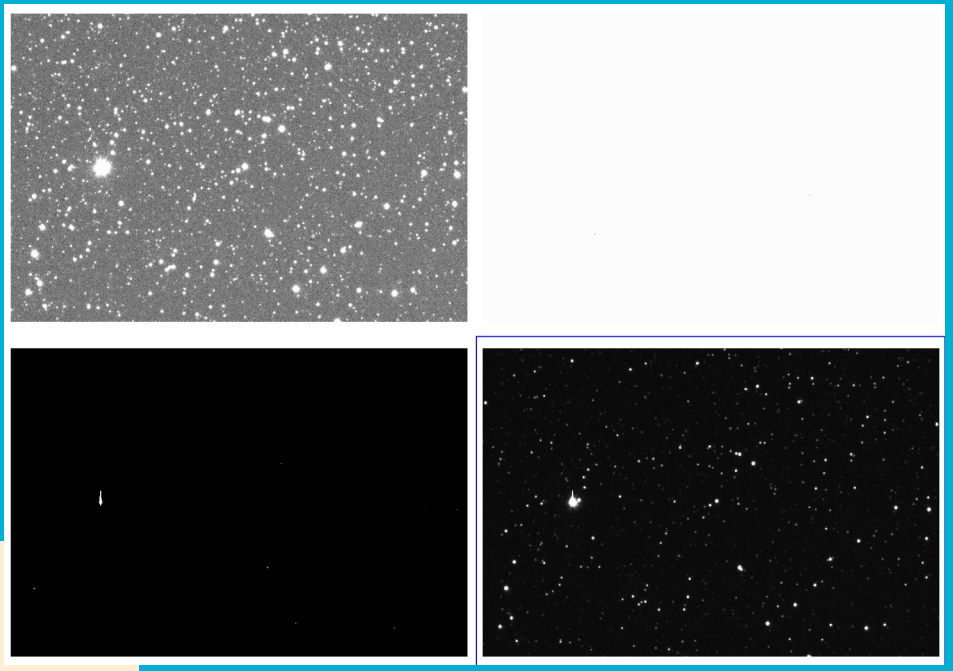
- Always have a copy of original raw data!

- **!mkdir reduction**
- **!mkdir reduction_copy**

- All the files that we work with have to be in the same directory:
  - science frames
  - masterflats (same filters as your science frames!)
  - darks

Check your data

epar display



```
                                Image Reduction and Analysis Facility
PACKAGE = tv
   TASK = display

image    =                          image to be displayed
frame    =                        1 frame to be written into
(bpmask =                      BPM) bad pixel mask
(bpdispl=                     none) bad pixel display (none|overlay|interpolate)
(bpcolor=                      red) bad pixel colors
(overlay=                         ) overlay mask
(ocolors=                    green) overlay colors
(erase   =                     yes) erase frame
(border_=                       no) erase unfilled area of window
(select_=                      yes) display frame being loaded
(repeat =                       no) repeat previous display parameters
(fill    =                      no) scale image to fit display window
(zscale =                      yes) display range of greylevels near median
(contras=                     0.25) contrast adjustment for zscale algorithm
(zrange =                      yes) display full image intensity range
(zmask  =                         ) sample mask
(nsample=                     1000) maximum number of sample pixels to use
(xcenter=                      0.5) display window horizontal center
(ycenter=                      0.5) display window vertical center
(xsize   =                      1.) display window horizontal size
(ysize   =                      1.) display window vertical size
(xmag    =                      1.) display window horizontal magnification
(ymag    =                      1.) display window vertical magnification
(order   =                       0) spatial interpolator order (0=replicate, 1=linear)
(z1      =                         ) minimum greylevel to be displayed
(z2      =                         ) maximum greylevel to be displayed
(ztrans =                      log) greylevel transformation (linear|log|none|user)
(lutfile=                         ) file containing user defined look up table
(mode    =                      ql)
```

Play around with these default values to get an image you're happy with

display (image name)

Check (all) your data

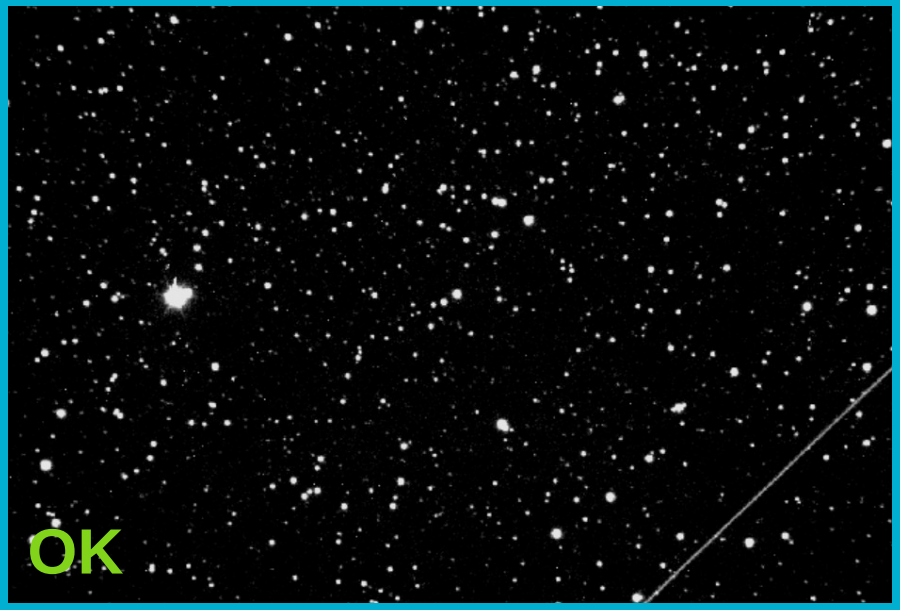
- All data is already in the same directory

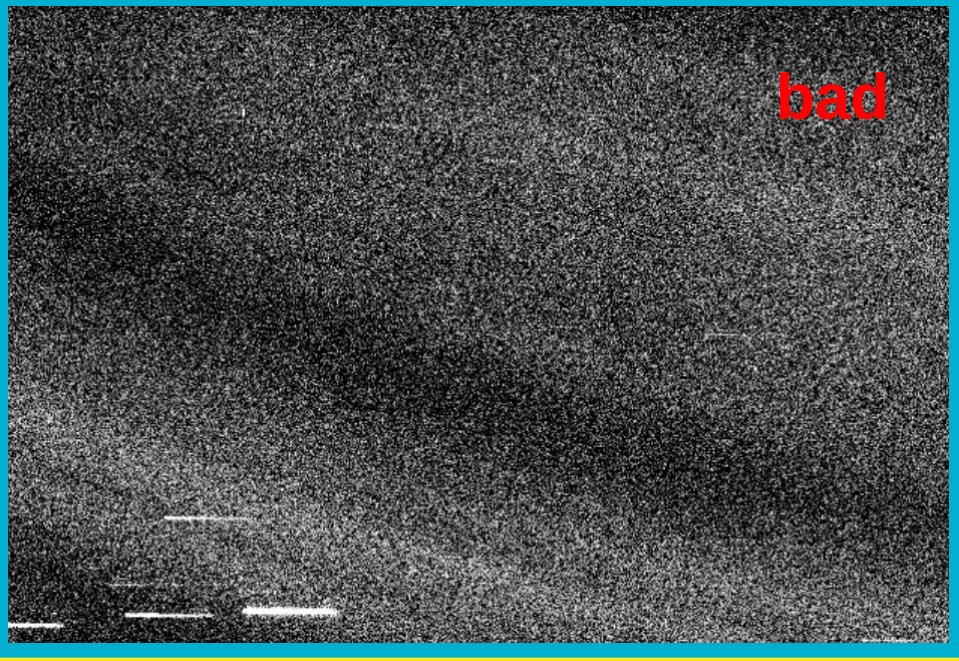
  !ds9 &

- Create a list with science frames.
  ls filename*.fits > list_science

- Display images in ds9 and relocate useless frames
  imexam @list_science 1
  (n next frame
  p previous frame
  q quit)

- Remove useless science frames: clouds? Satellites? tracking problems? Etc.
  But avoid removing unnecessarily!!


OK


bad

Fixing the fits headers

- Add the following to ALL fits headers (new camera)

hedit *.fit BIASSEC '[1058:1062,2:1022]' add+ ver-
hedit *.fit TRIMSEC '[1:1056,1:1026]' add+ ver-
hedit *.fit datasec del+ ver-

- Move the darks and flats into separate directories (if needed)

- Change the image type in the science images to ‘object’

hedit *fit imagetyp 'object' add+ ver-

Creating a master dark

- What is the exposure time of the images we will analyse?
  Check the header!
  Single frame: **imhead [image name] lo+ | page**
  Multible frames: **hsel Object*fits $I,exptime yes**
  (“exptime” is the fits header keword)

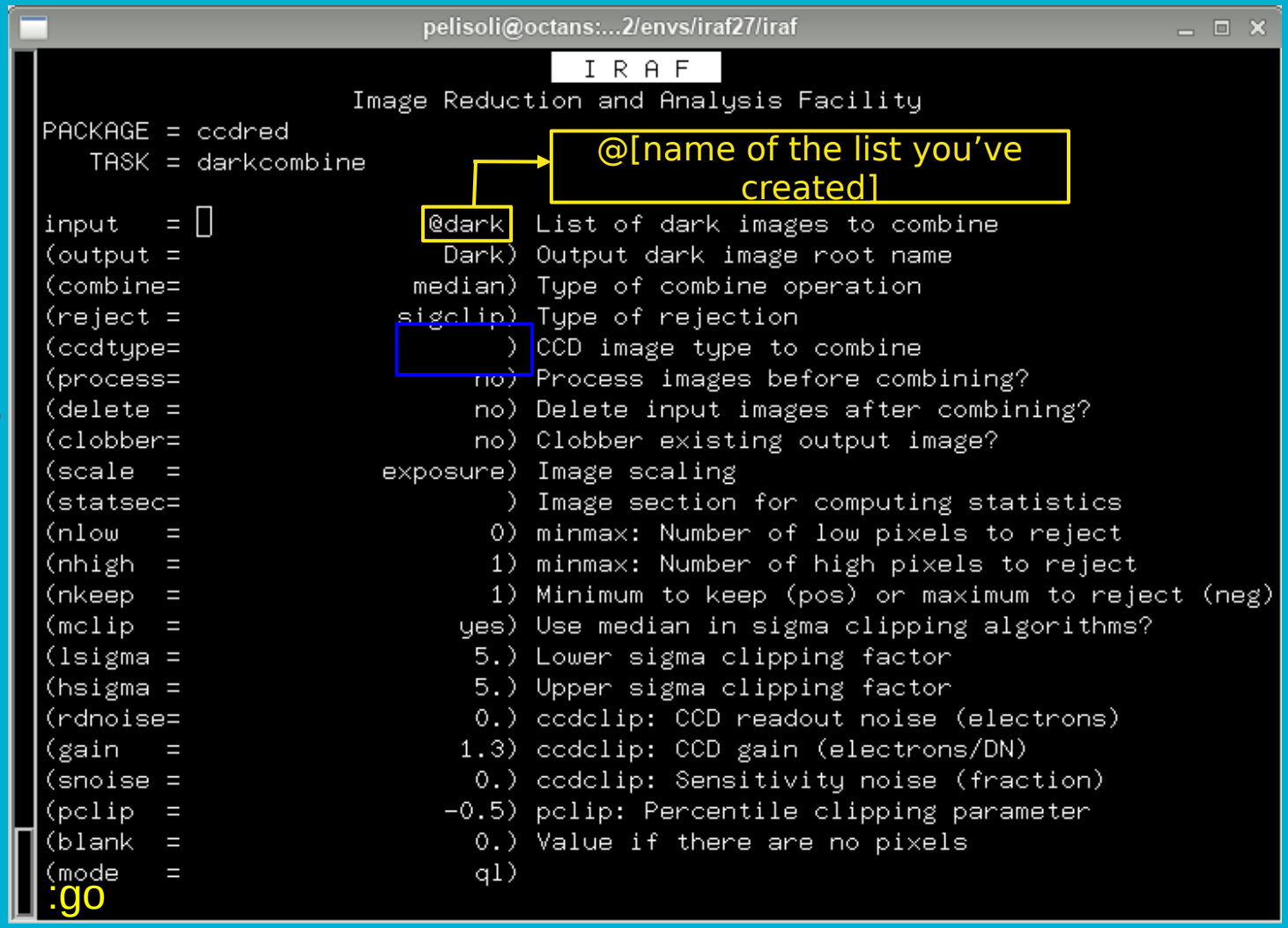
- Which dark images should we use?
  imhead df–* lo+ | grep EXPTIME or **hsel**
- Create a list (text document) containing the names of the dark frames using the same exposure time as the science images.

  Ls df* > df_60s_list

Creating a master dark – epar darkcombine

Check parameters!

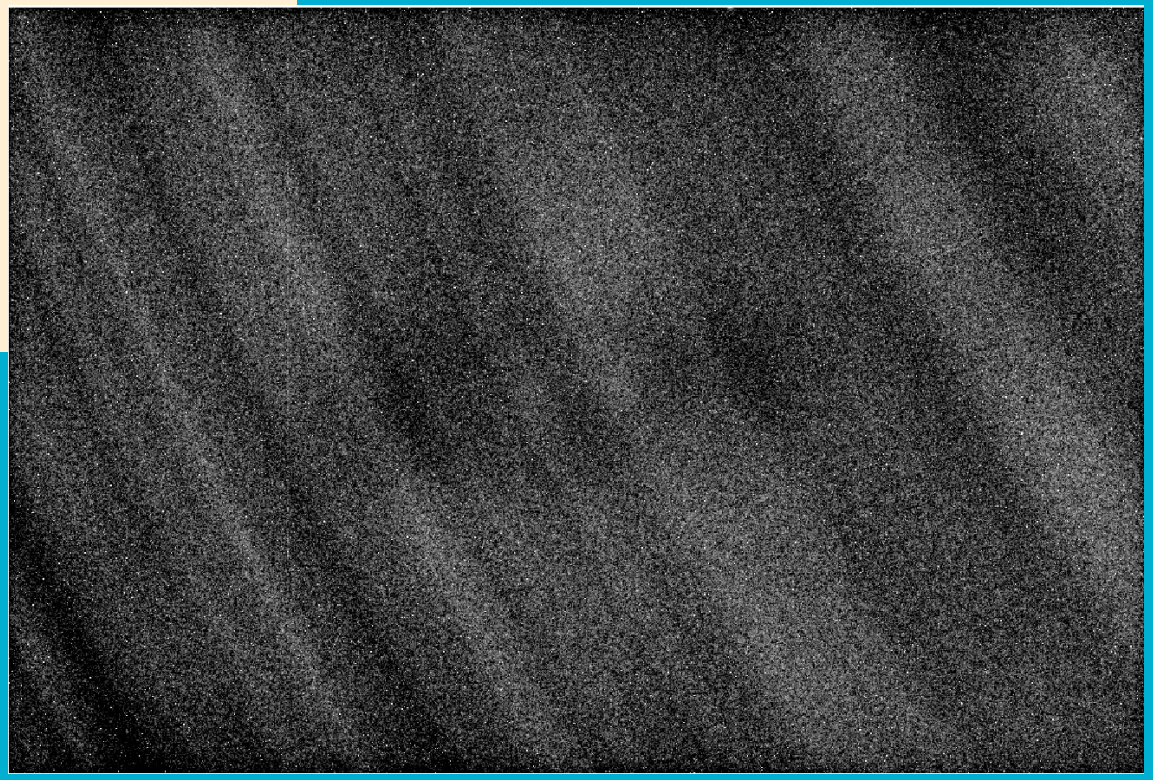
ccdtype has to be empty! Just put space.



":go" + enter

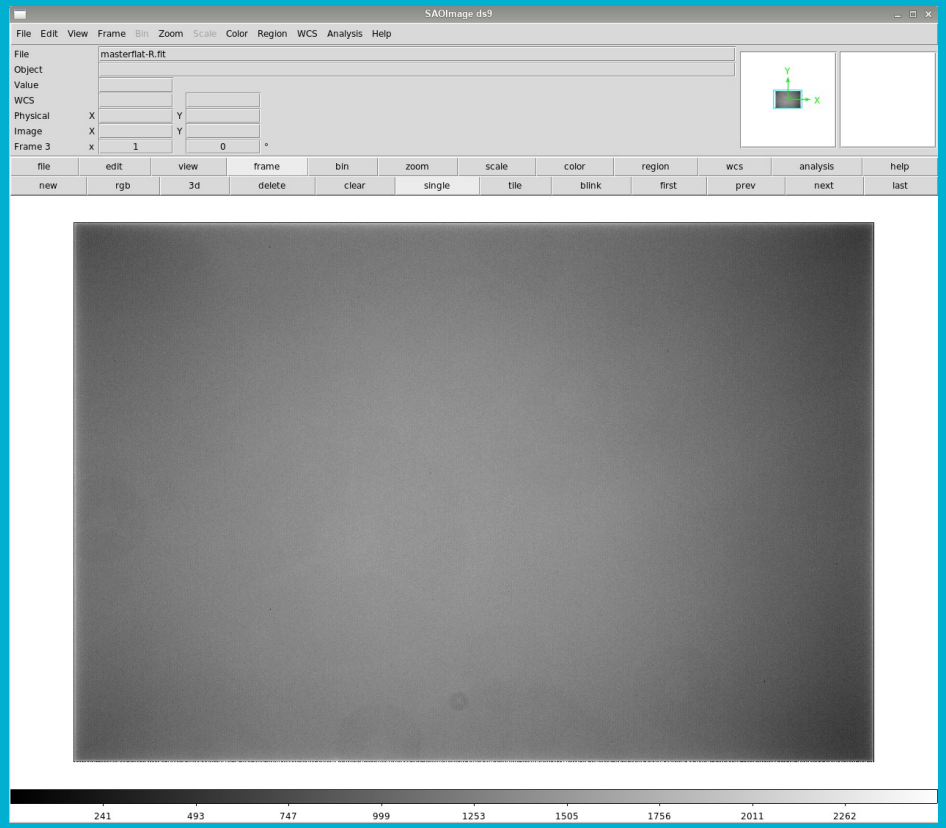
Creating a master dark –

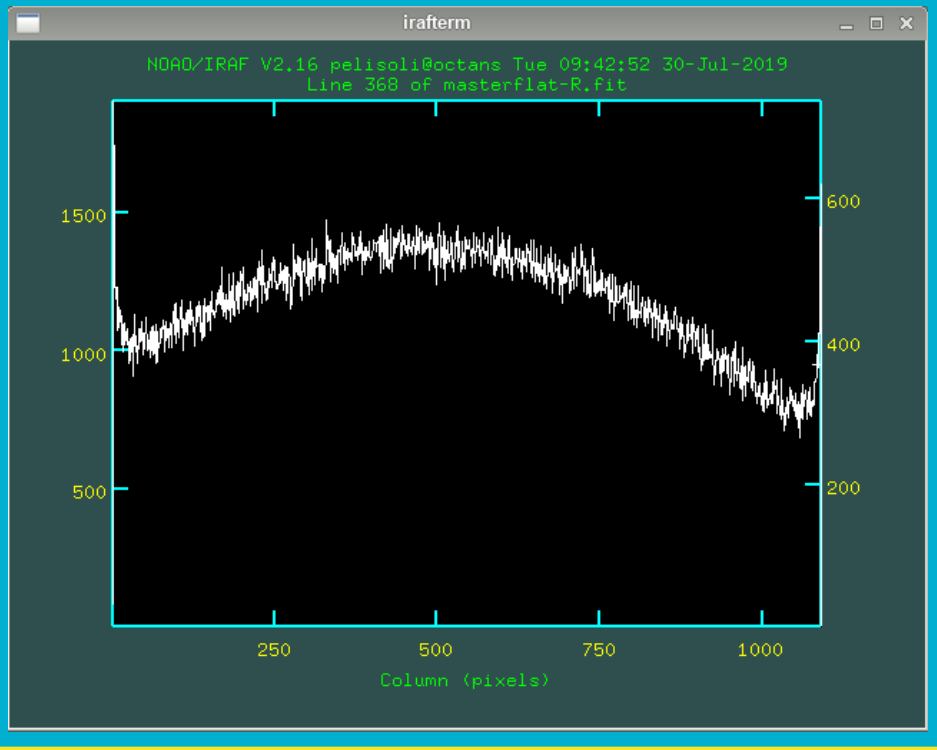
```
Aug 30 16:01: IMCOMBINE
  combine = median, scale = exposure, zero = none, weight = none
  reject = sigclip, mclip = yes, nkeep = 1
  lsigma = 5., hsigma = 5.
  blank = 0.
                Images
        df60s__0001.fits
        df60s__0002.fits
        df60s__0003.fits
        df60s__0004.fits
        df60s__0005.fits
        df60s__0006.fits
        df60s__0007.fits
        df60s__0008.fits
        df60s__0009.fits
        df60s__0010.fits

  Output image = Dark, ncombine = 10
```



- Check what you got!
  display
  imstat

Master flat





Remember to subtract your masterdark!

Reducing the science images

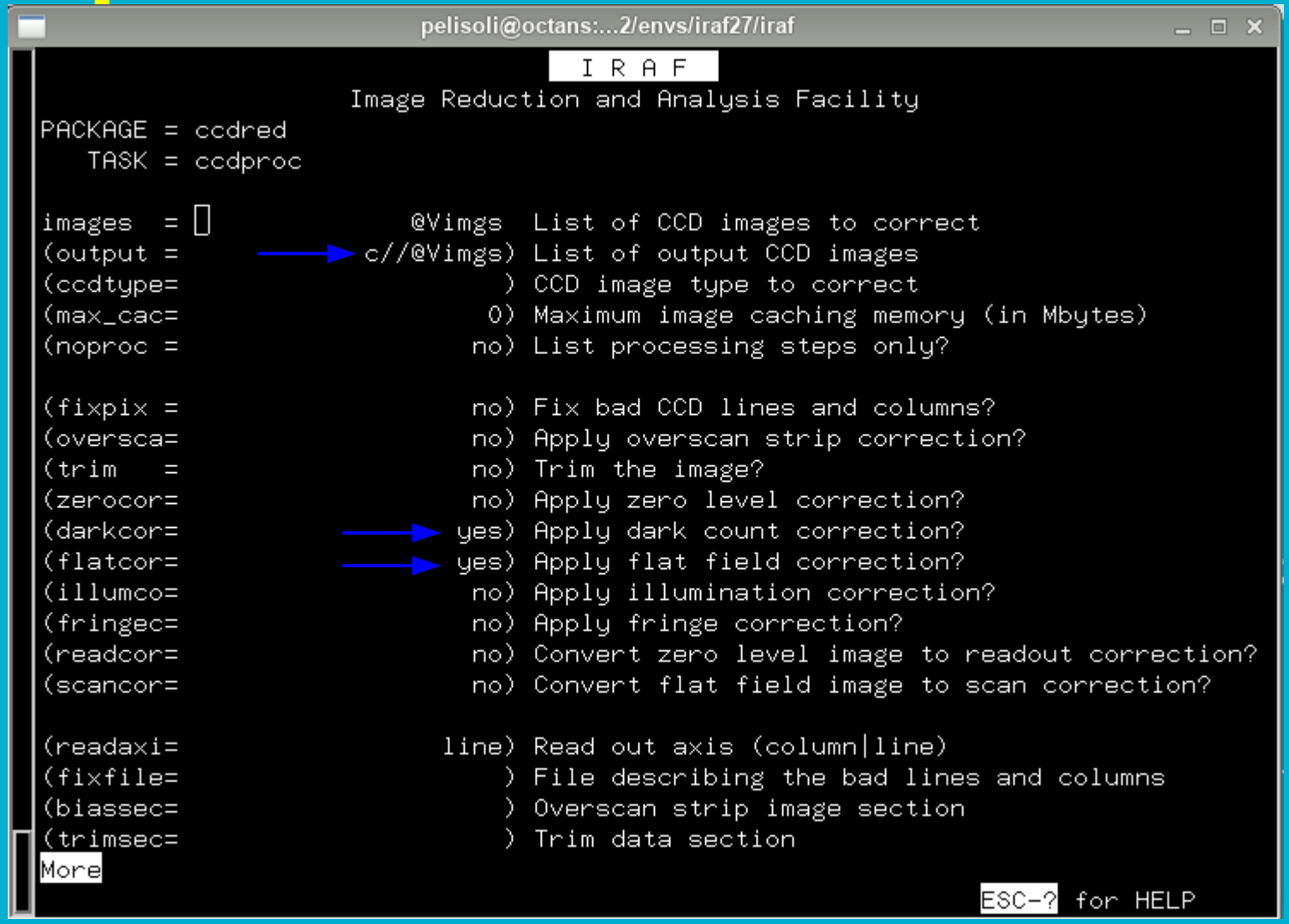
- We likely have images of different filters: e.g. R and V.
  You have to use the correct master flat for each of them.
- Make a list containing the R images, and another containing the V images, e.g.

  ls Cyg2*R*.fit > Rimgs
  ls Cyg2*V*.fit > Vimgs

- Use the task **ccdproc** to divide the images by the flat and subtract the dark current. Do it separately for R and V images.

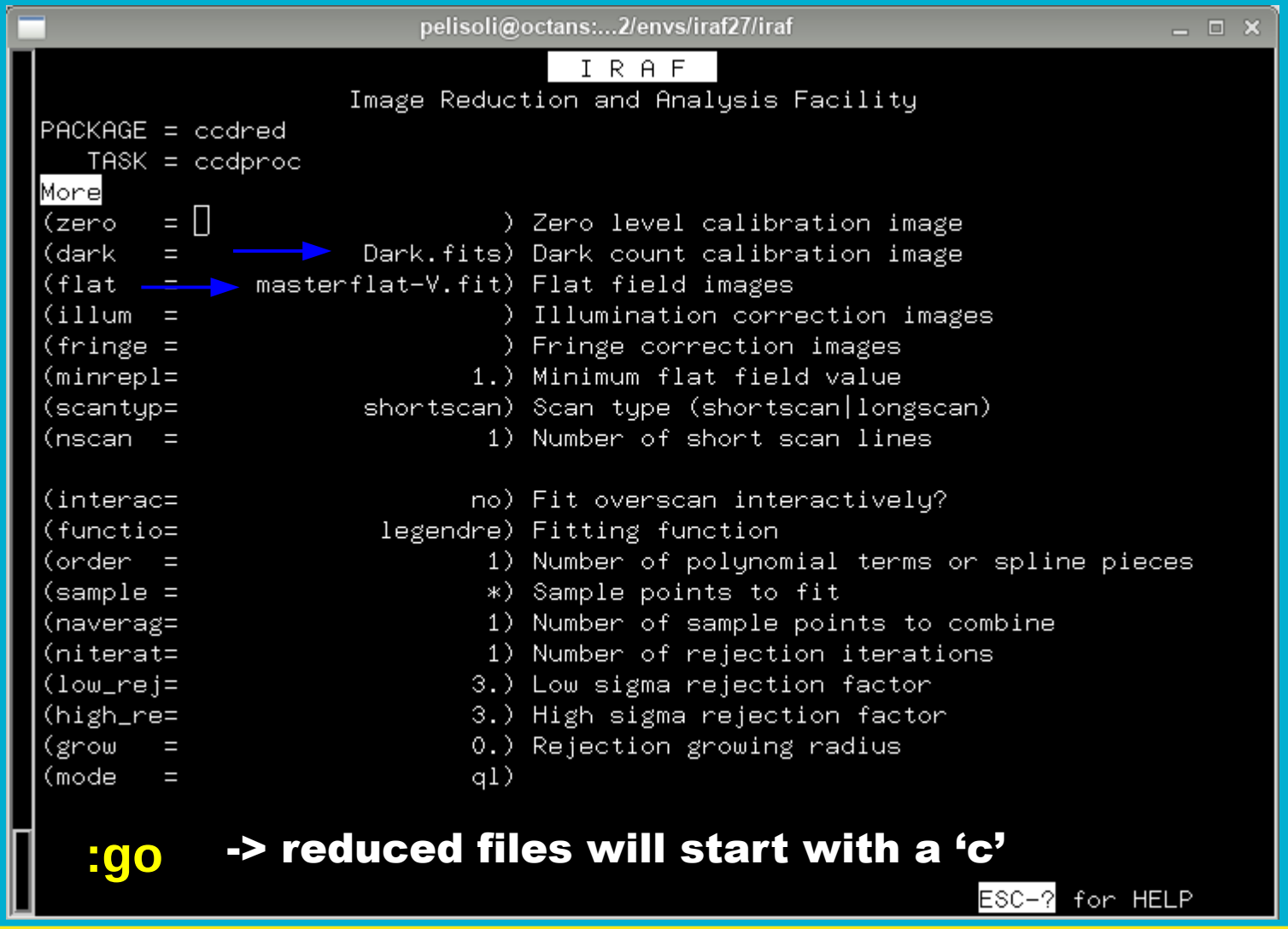
Reducing the science images 1 - epar ccdproc

Check parameters!



Continues ..

Reducing the science images 2 - epar ccdproc



Photometry

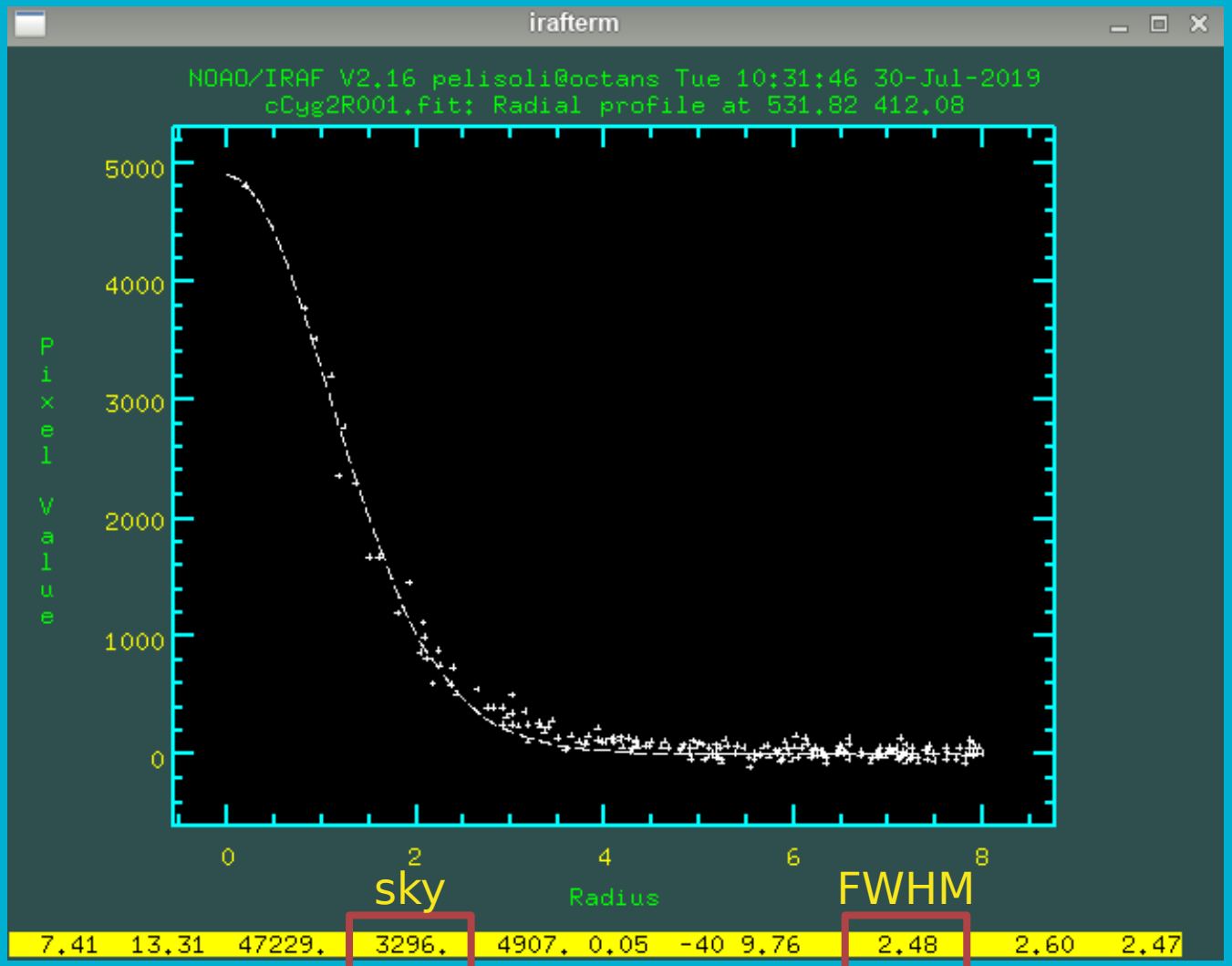
- Now that the images have been reduced, we can perform photometry.
- The first step is to run the task **daofind**, which will find the stars in our images.
- There are a few parameters we need to measure in our image to best setup daofind: the sky and the **F**(ull)**W**(idth)**H**(alf)**M**(aximum)
- For that, display an image at the beginning of the sequence, middle, and end:

display cCyg2R001.fit 1
display cCyg2R111.fit 2
display cCyg2R223.fit 3

- Use the task **imexamine** – choose a relatively bright star near the centre of the image. Centre the cursor on this star.

r → display the radial profile
e → show contours
a → write measurements to the screen

Photometry


irafterm
NOAO/IRAF V2.16 pelisoli@octans Tue 10:31:46 30-Jul-2019
cCyg2R001.fit: Radial profile at 531.82 412.08
Pixel Value
5000
4000
3000
2000
1000
0
0
2
4
6
8
Radius
sky
FWHM
7.41 13.31 47229. 3296. 4907. 0.05 -40 9.76 2.48 2.60 2.47

Photometry

- Check the sky values in the three images. We will use this to set our initial guess for the background. The value of sigma is in turn the square-root of the background (assuming Poissonic noise).

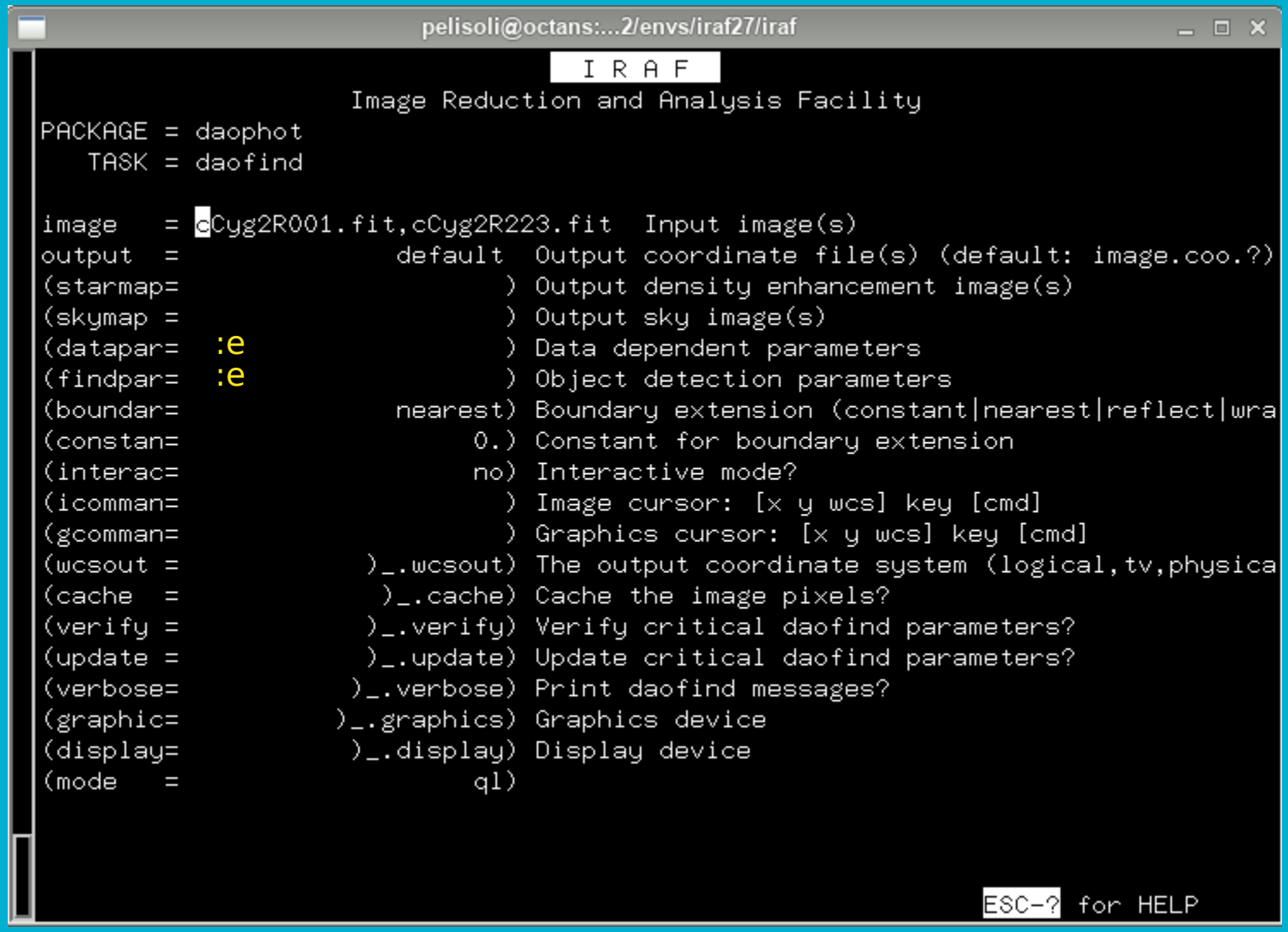
  If the values are very different, use the median; if they are similar, use the mean.

  E. g.

  sky = 415.
  sigma = 20.4

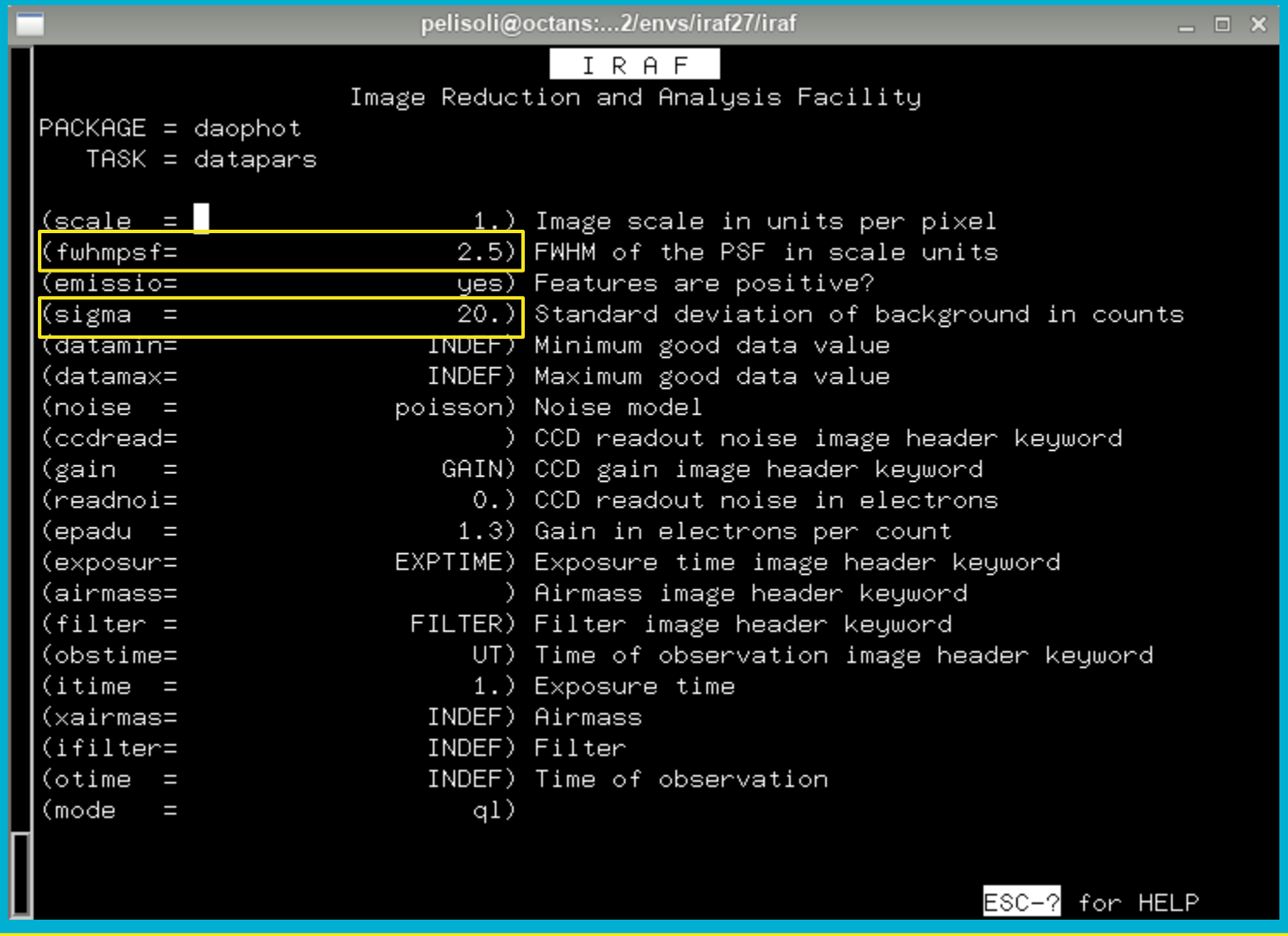
- Check the FWHM in the three images. We will use this to set the aperture and the sky region for the photometry.

epar DAOFIND (to find the stars coordinates)

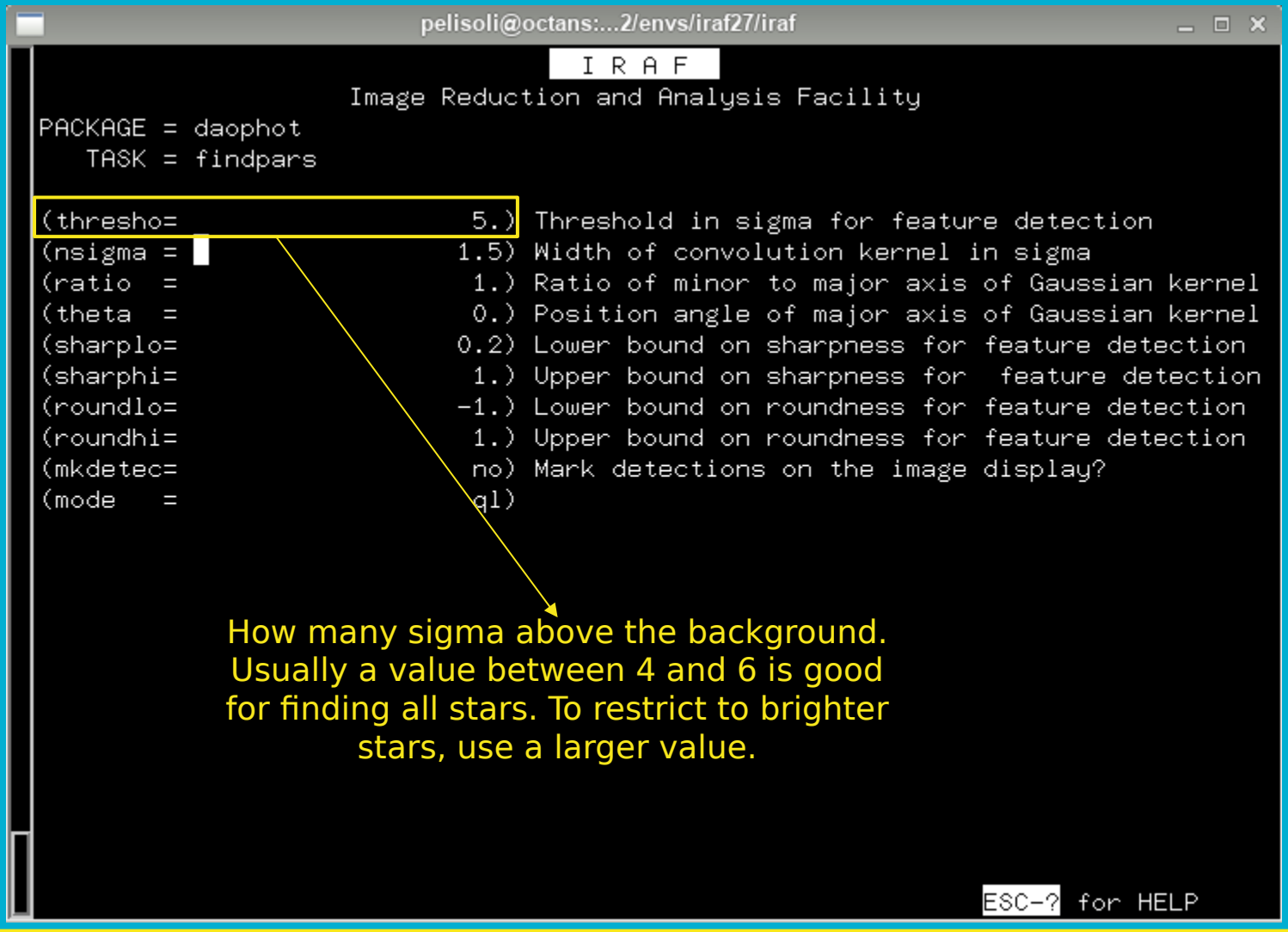


DAOFIND (datapars)

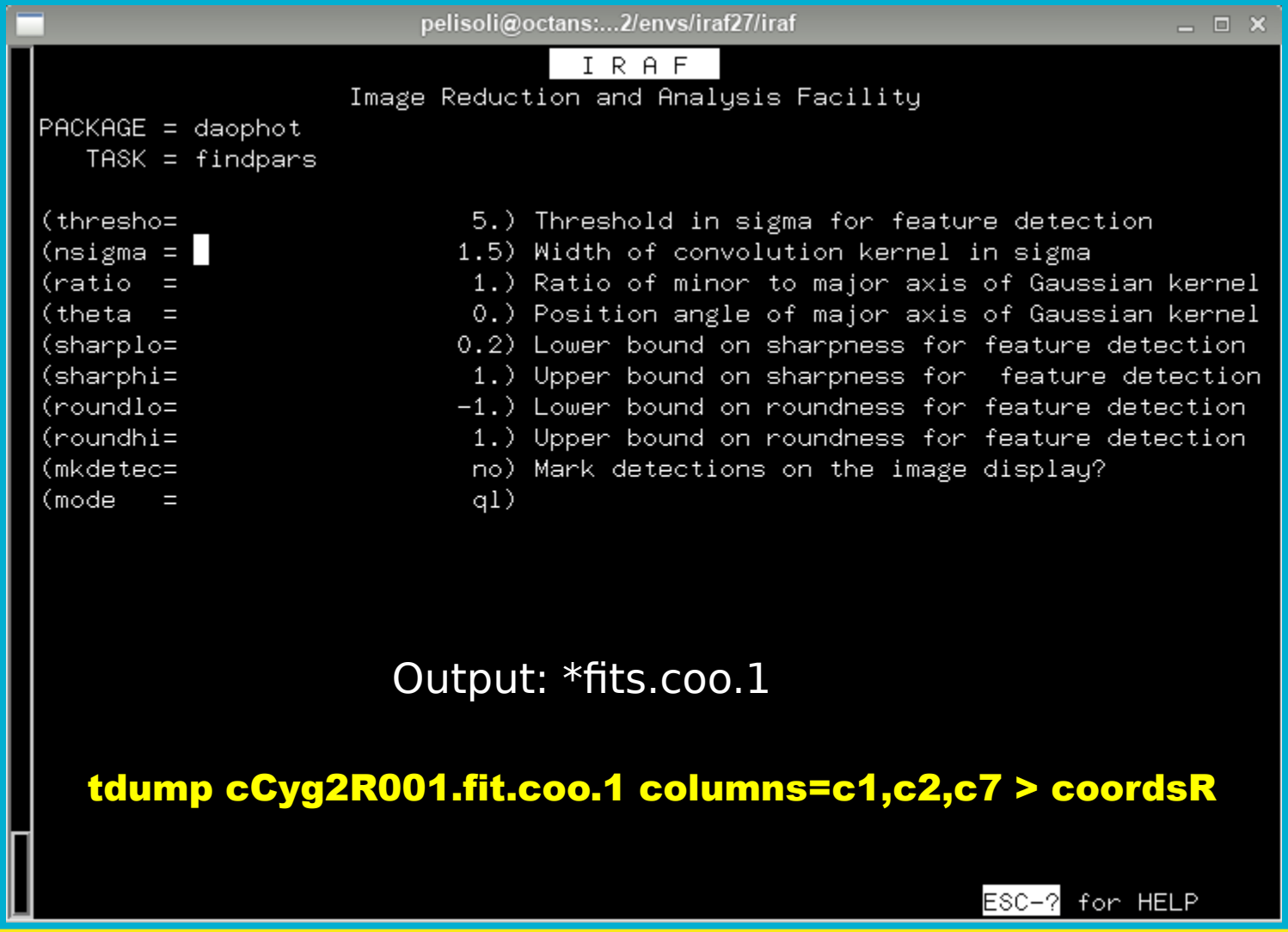
"**:q**" to go back



DAOFIND (findpars)



DAOFIND (findpars)

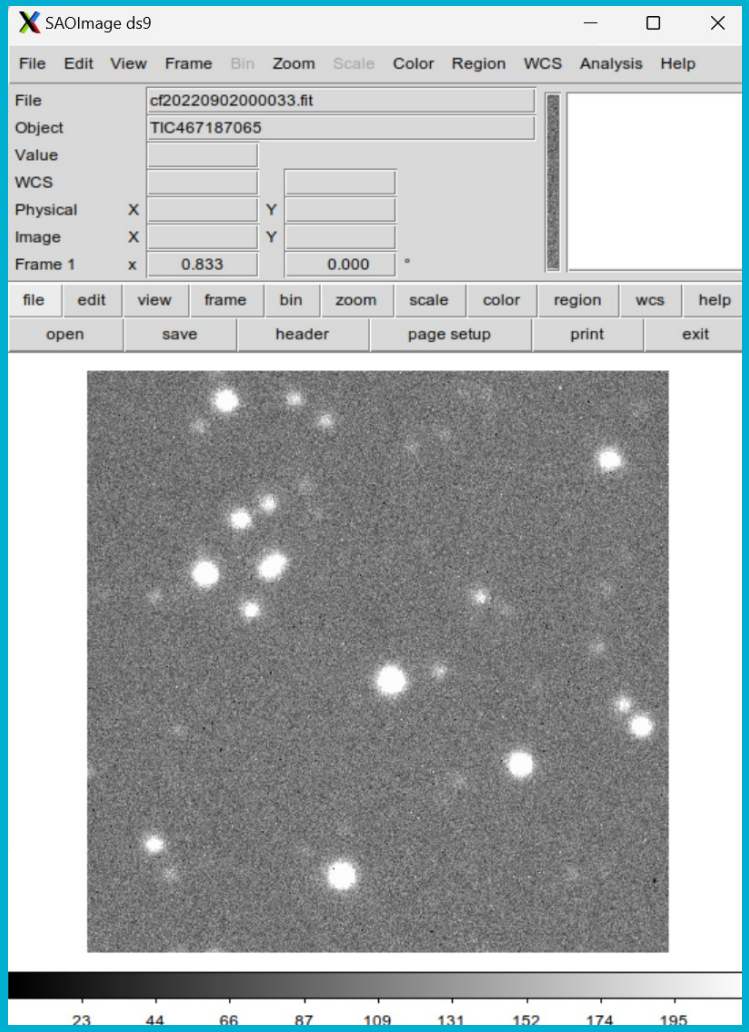


Output: *fits.coo.1

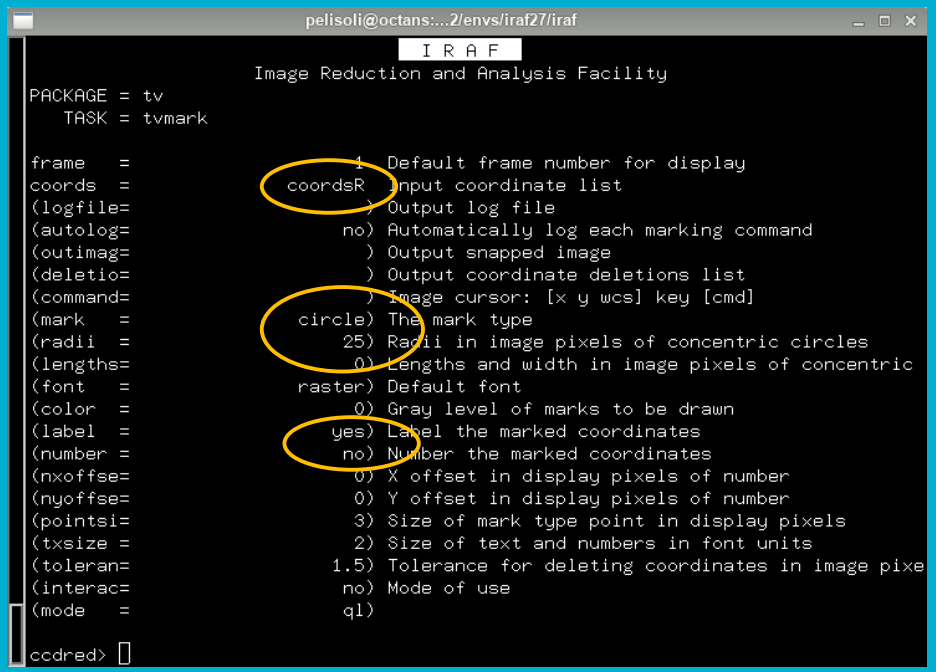
tdump cCyg2R001.fit.coo.1 columns=c1,c2,c7 > coordsR

DAOFIND (tvmark: to display the found stars)

display cObject...

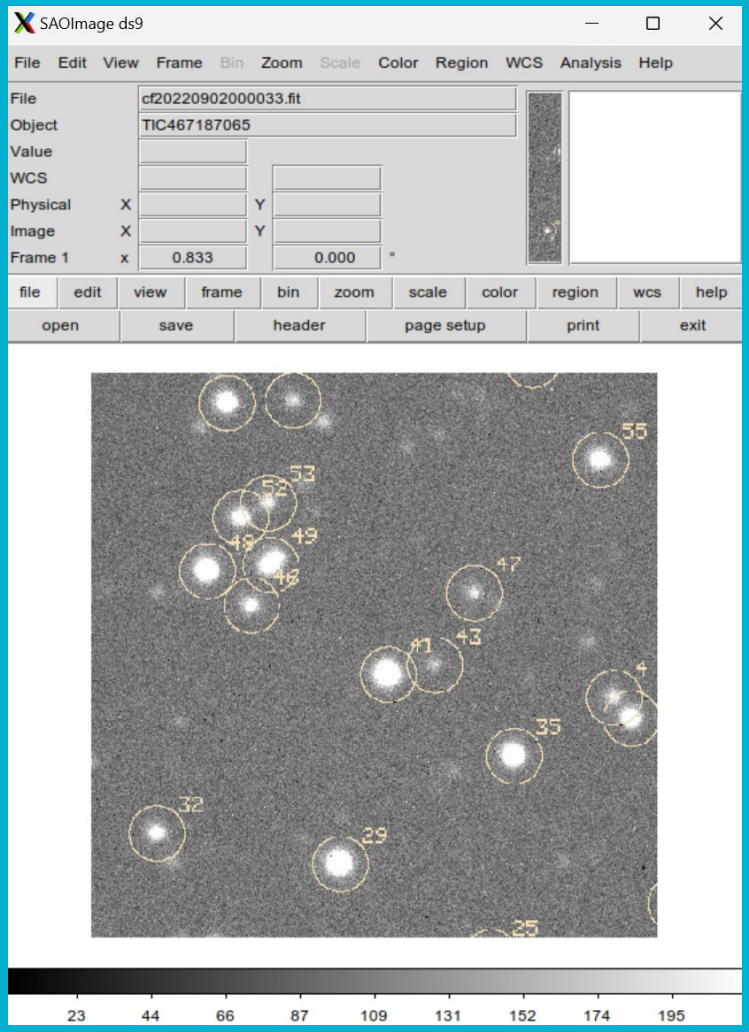


epar tvmark
:go

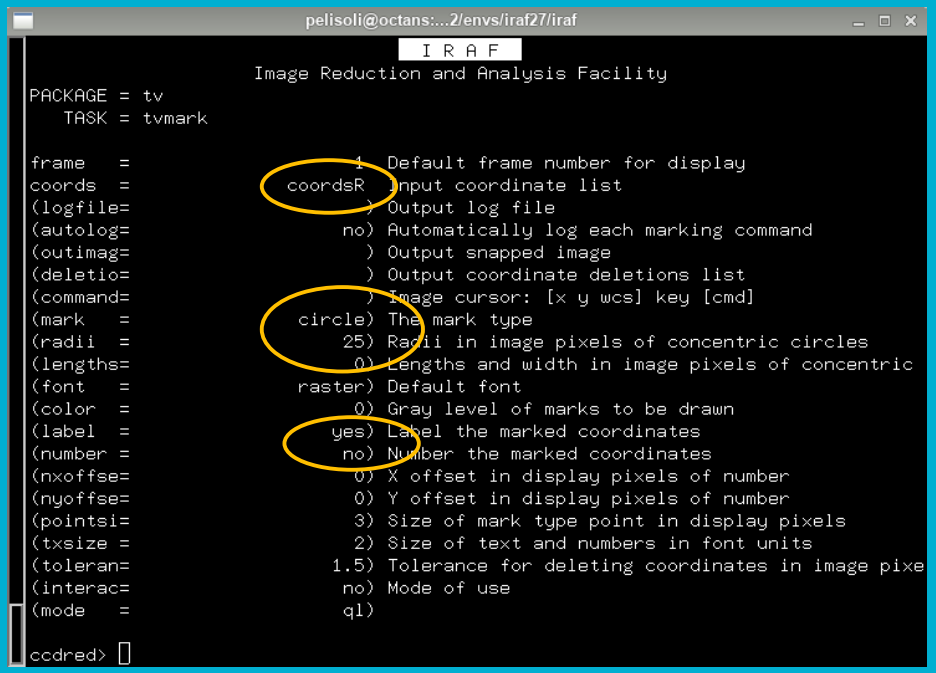


DAOFIND (tvmark: to display the found stars)

display cObject...



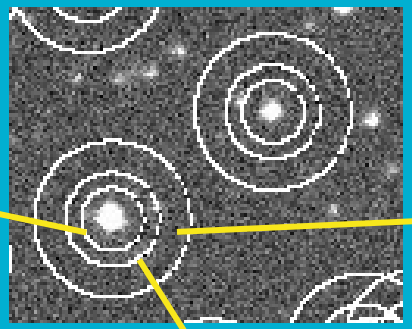
epar tvmark
:go



Photometry

- tvmark is also useful to help us define the aperture, annulus, and dannulus

Aperture (radii): where the flux of the star will be measured.
Usually ~2.5 x FWHM



Dannulus: width of the ring to count the background.
~5-10 pixels

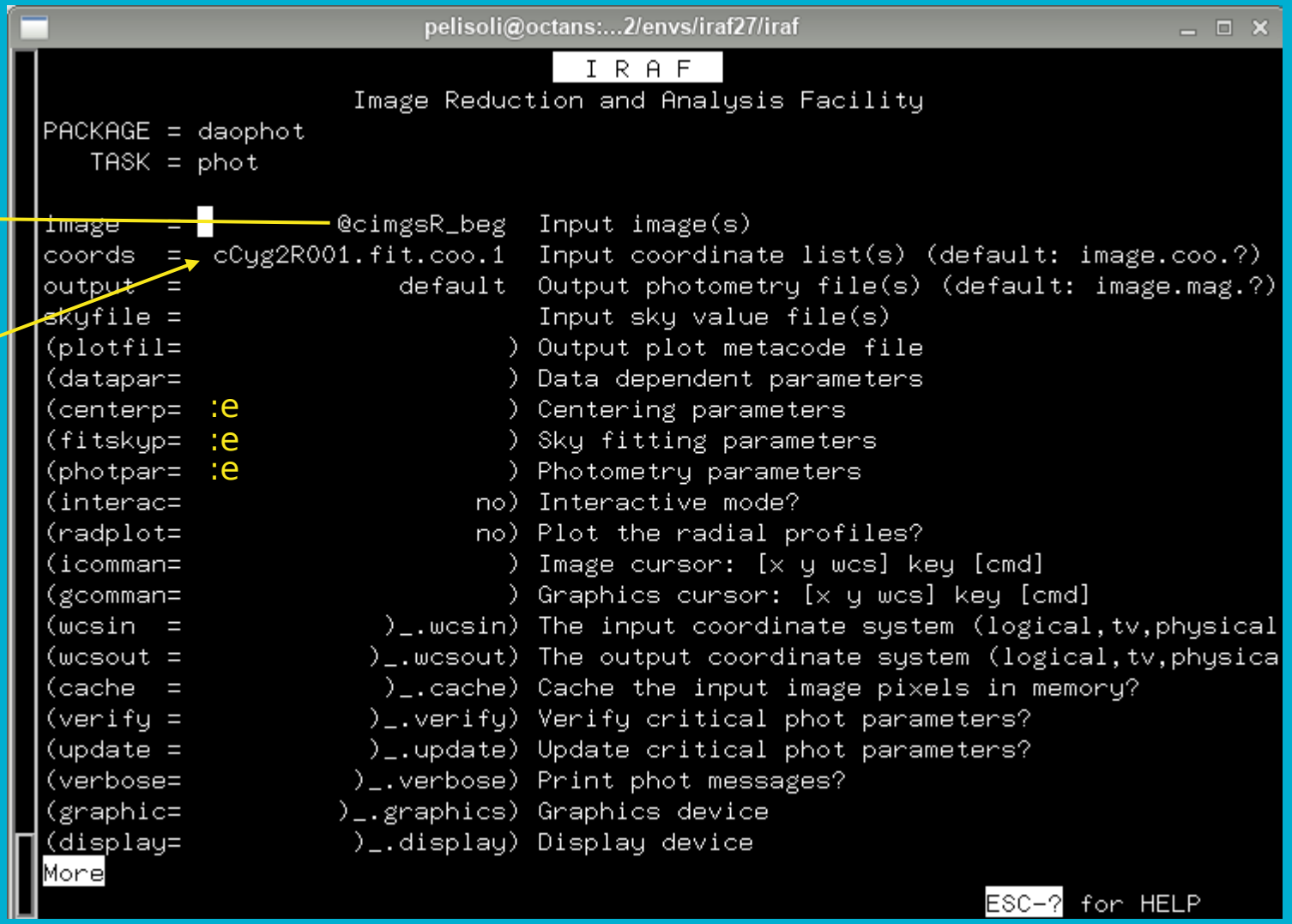
Annulus (radii): distance at which to start counting the background.
At least 2.5 x FWHM
~4 x FWHM in our example

* For a Gaussian distribution:
FWHM = 2.35σ
99.99% of the light is contained within
4σ = 1.7FWHM

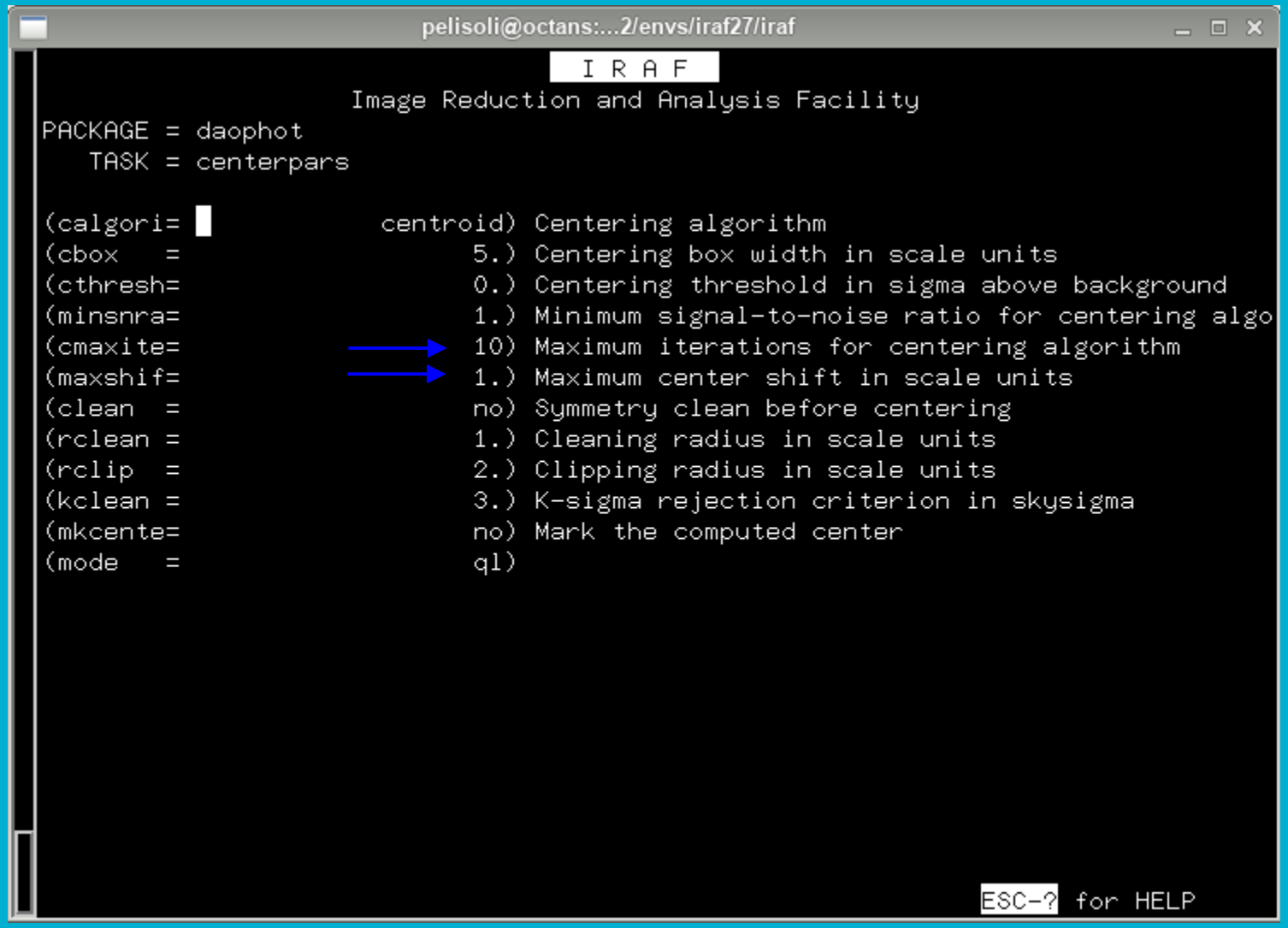
DAOPHOT – epar phot

List of reduced R (or V/I/R) images.

NB!
One coordinate file for all frames

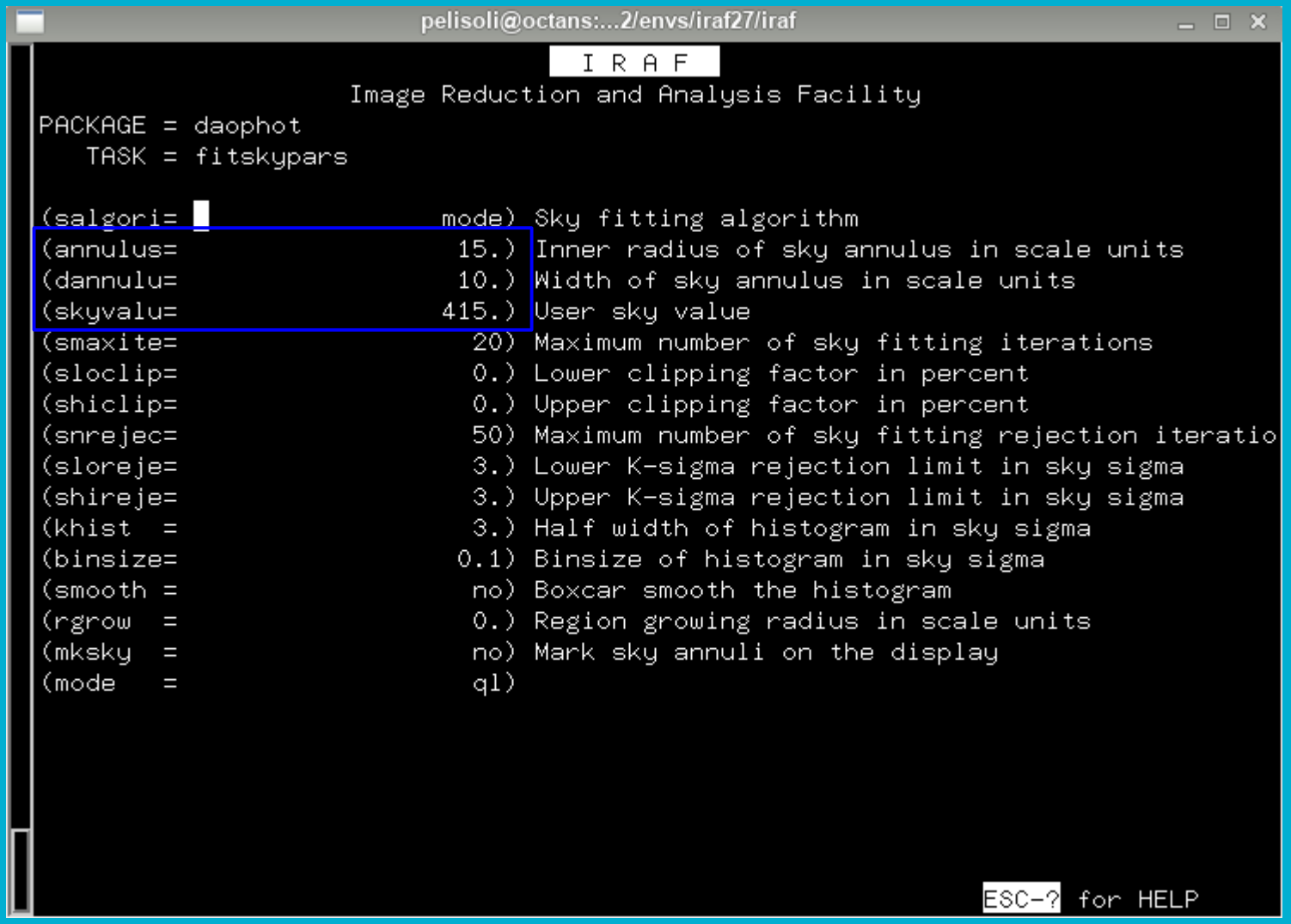


DAOPHOT (centerpars)

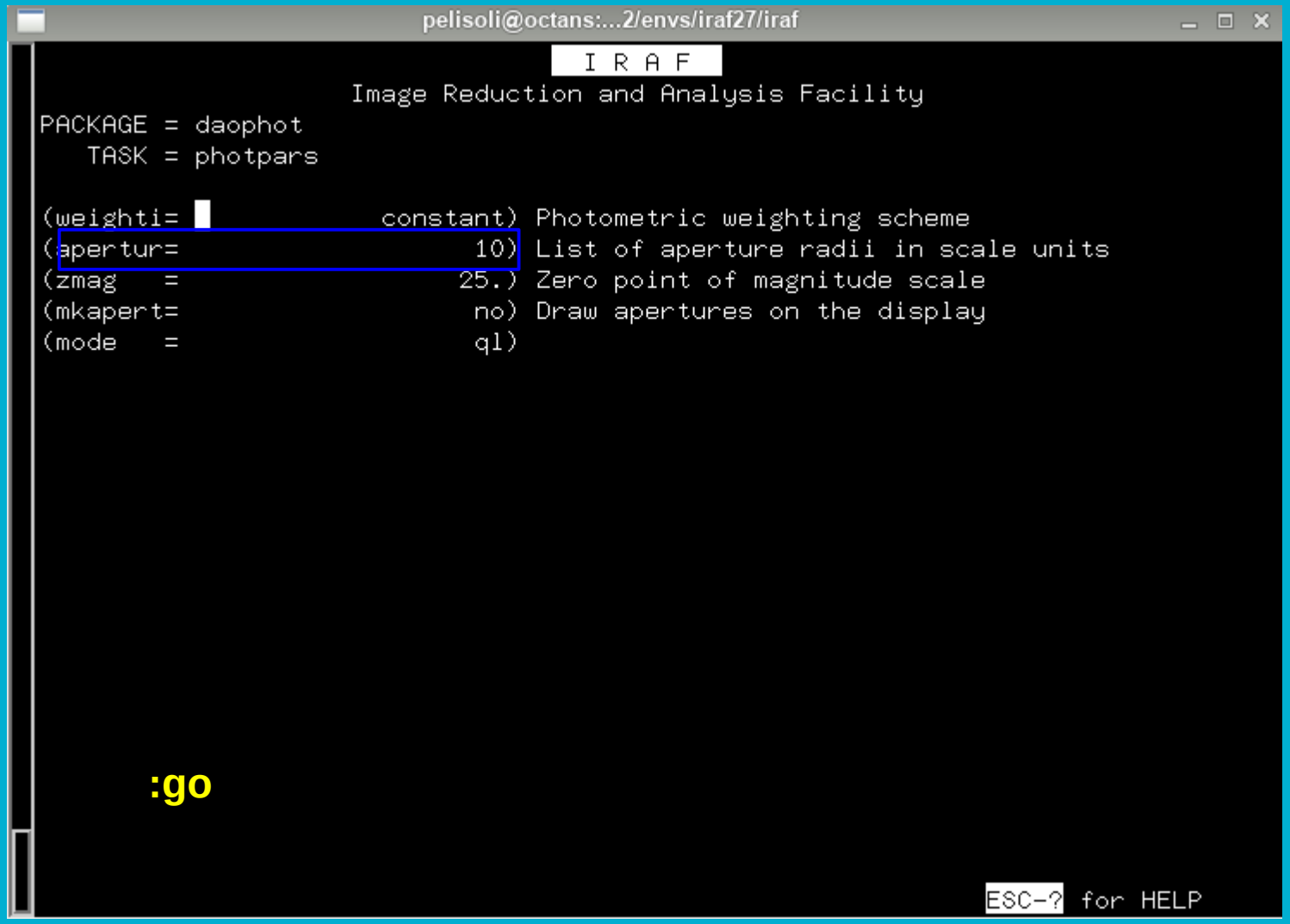


PHOT (fitskypars)

Your data specific!

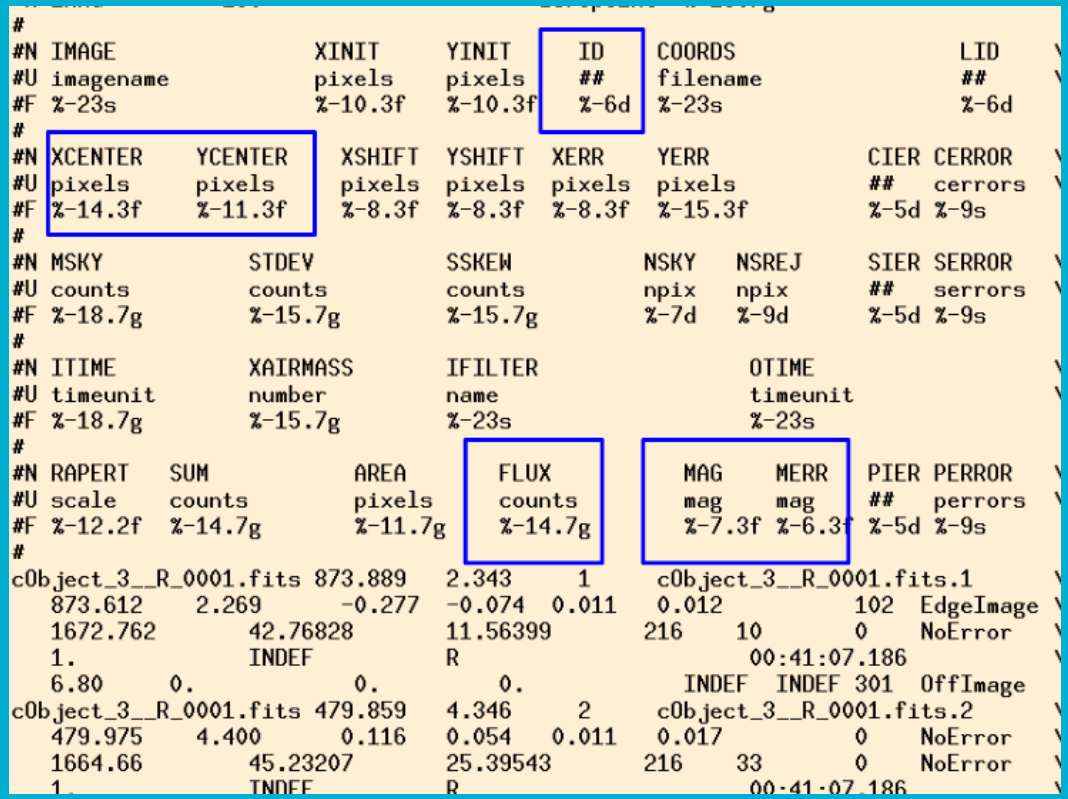


PHOT


pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = photpars
(weighti= constant) Photometric weighting scheme
(aperture= 10) List of aperture radii in scale units
(zmag = 25.) Zero point of magnitude scale
(mkapert= no) Draw apertures on the display
(mode = ql)
:go
ESC-? for HELP

PHOT
(outputs)

- Text files:
  ***.fits.coo.1.mag.1**
  or ***mag.2** if done twice



```
#
#N IMAGE                XINIT     YINIT     ID    COORDS                  LID     \
#U imagename            pixels    pixels    ##    filename                ##      \
#F %-23s                %-10.3f   %-10.3f   %-6d  %-23s                   %-6d
#
#N XCENTER    YCENTER   XSHIFT    YSHIFT    XERR     YERR          CIER CERROR   \
#U pixels     pixels    pixels    pixels    pixels   pixels        ##   cerrors  \
#F %-14.3f    %-11.3f   %-8.3f    %-8.3f    %-8.3f   %-15.3f       %-5d %-9s
#
#N MSKY          STDEV          SSKEW          NSKY   NSREJ     SIER SERROR   \
#U counts        counts         counts         npix   npix      ##   serrors  \
#F %-18.7g       %-15.7g        %-15.7g        %-7d   %-9d      %-5d %-9s
#
#N ITIME         XAIRMASS       IFILTER                 OTIME                 \
#U timeunit      number         name                    timeunit              \
#F %-18.7g       %-15.7g        %-23s                   %-23s
#
#N RAPERT   SUM          AREA       FLUX       MAG     MERR     PIER PERROR   \
#U scale    counts       pixels     counts     mag     mag      ##   perrors  \
#F %-12.2f  %-14.7g      %-11.7g    %-14.7g    %-7.3f  %-6.3f   %-5d %-9s
#
cObject_3__R_0001.fits 873.889   2.343     1      cObject_3__R_0001.fits.1      \
   873.612    2.269     -0.277  -0.074  0.011   0.012          102  EdgeImage \
   1672.762        42.76828        11.56399        216    10       0    NoError \
   1.              INDEF           R                       00:41:07.186         \
   6.80       0.          0.          0.                   INDEF  INDEF 301  OffImage
cObject_3__R_0001.fits 479.859   4.346     2      cObject_3__R_0001.fits.2      \
   479.975    4.400     0.116   0.054   0.011   0.017          0    NoError \
   1664.66         45.23207        25.39543        216    33       0    NoError \
   1.              INDEF           R                       00:41:07.186         \
```

IMPORTANT COLUMNS
c4 = star ID
c5 = x coordinate
c6 = y coordinate
c29 = magnitude
c30 = magnitude error
c28= flux.

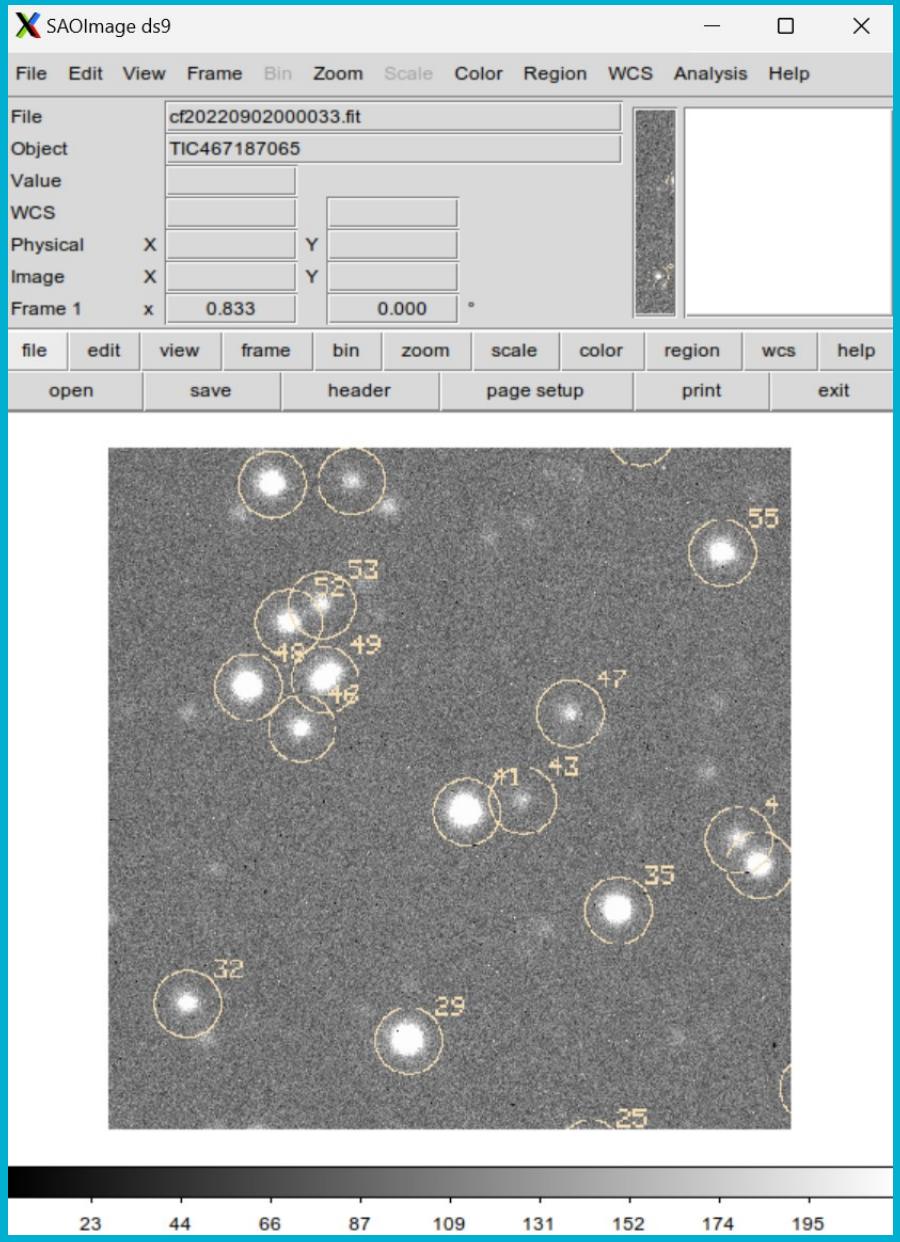
Make a list of photometry files:

ls *R*mag.1 > Rmag_files

txdump @Vmag_files fields=ID,XCENTER,YCENTER,FLUX,MAG,MERR > R_mags

Photometry

- Check the ID of your star and of a few comparison stars with tvmark. https://aladin.u-strasbg.fr/AladinLite/ might be useful to help identify your star.
- Comparison stars are needed to remove background variations from the light curve.



Photometry (more *massaging* of text files)

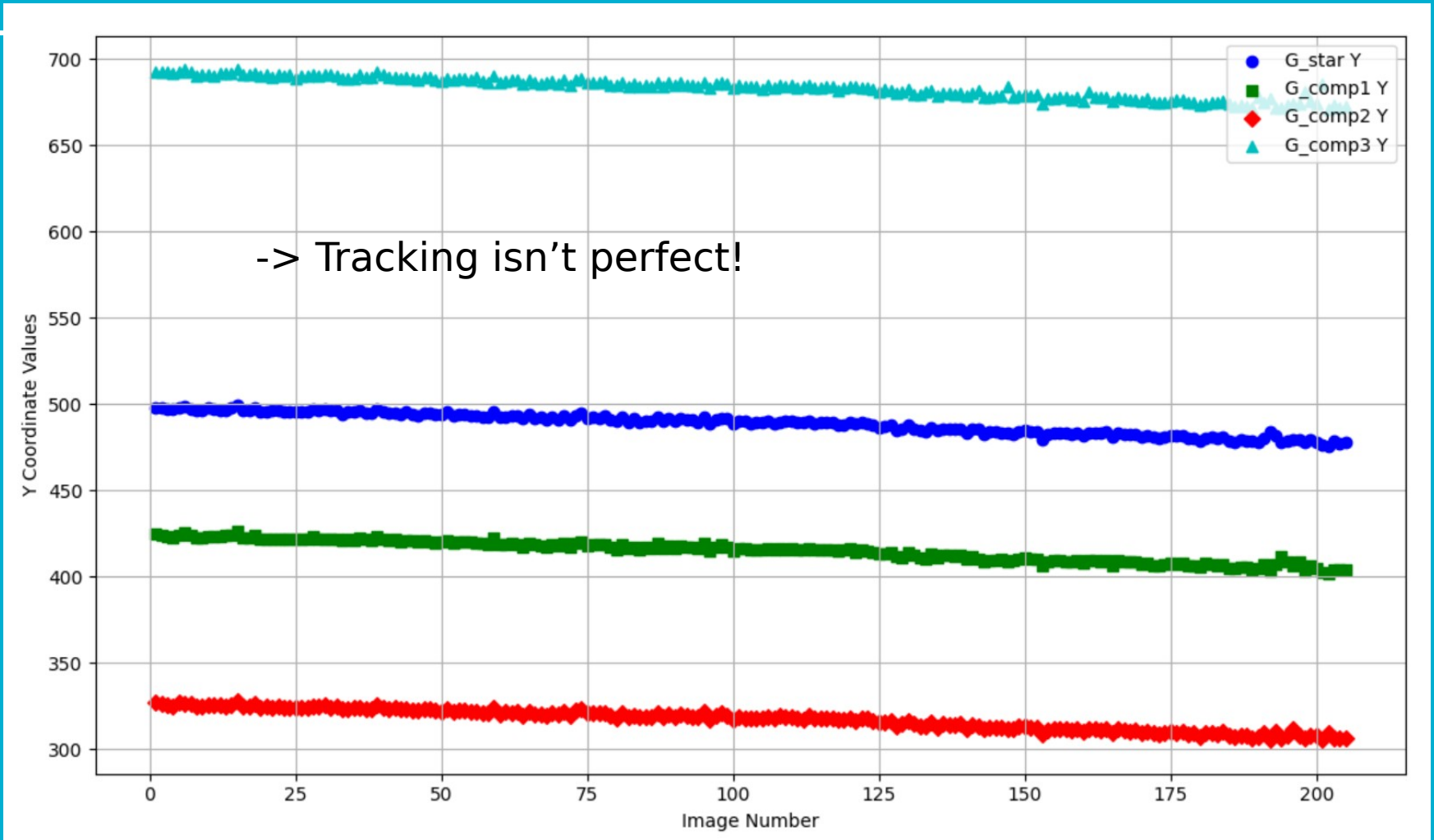
- Copy the photometry of the star and each comparison into separate files.

```
! awk '{if ($1==41) print;}' R_mags > R_star
! awk '{if ($1==43) print;}' R_mags > R_comp1
! awk '{if ($1==55) print;}' R_mags > R_comp2
! awk '{if ($1==29) print;}' R_mags > R_comp3
! awk '{if ($1==32) print;}' R_mags > R_comp4
```

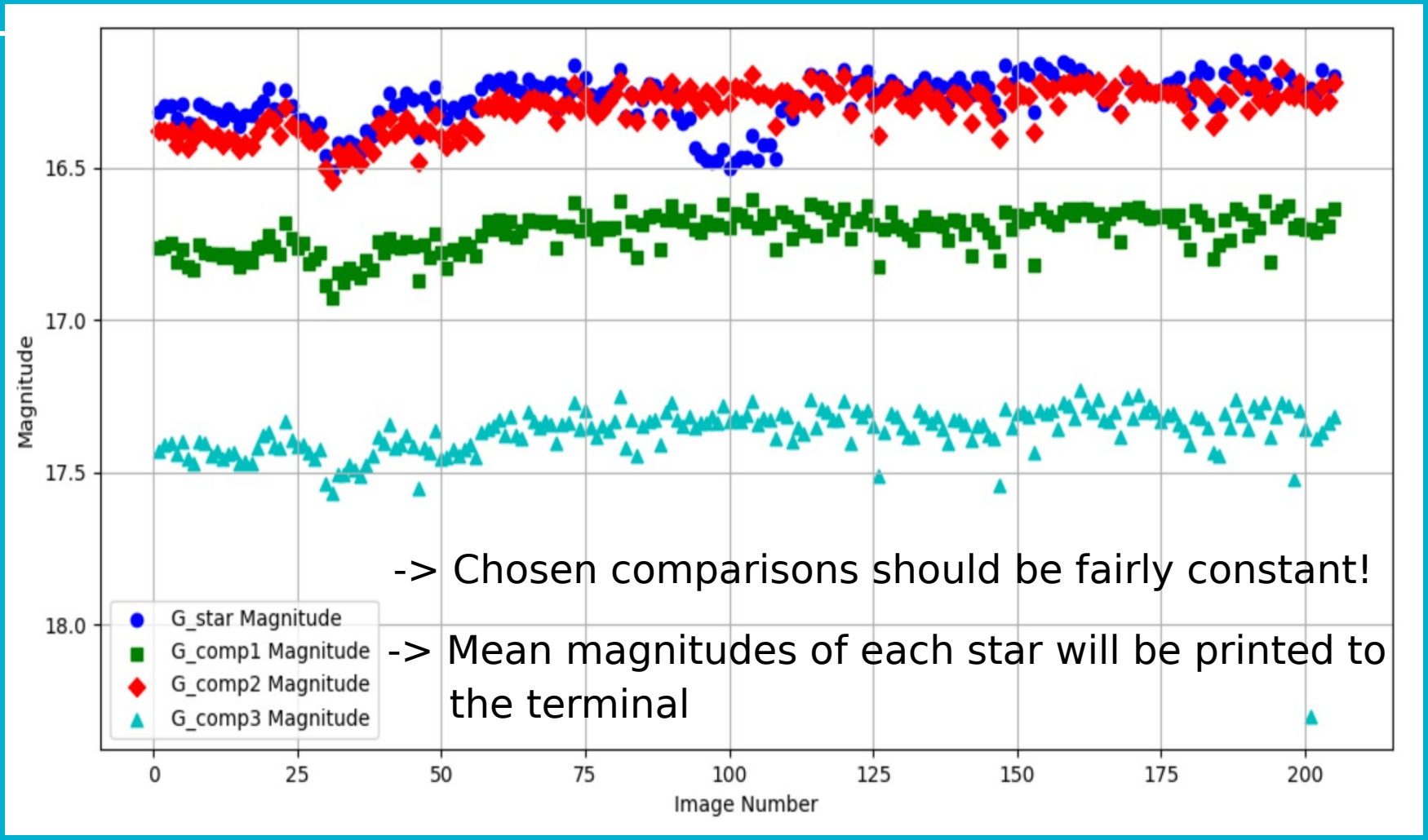
(Replace with correct Ids)

- Plot these files using the provided code: ‘plot_mag_files.py’

Photometry – shifting images



Photometry - magnitudes



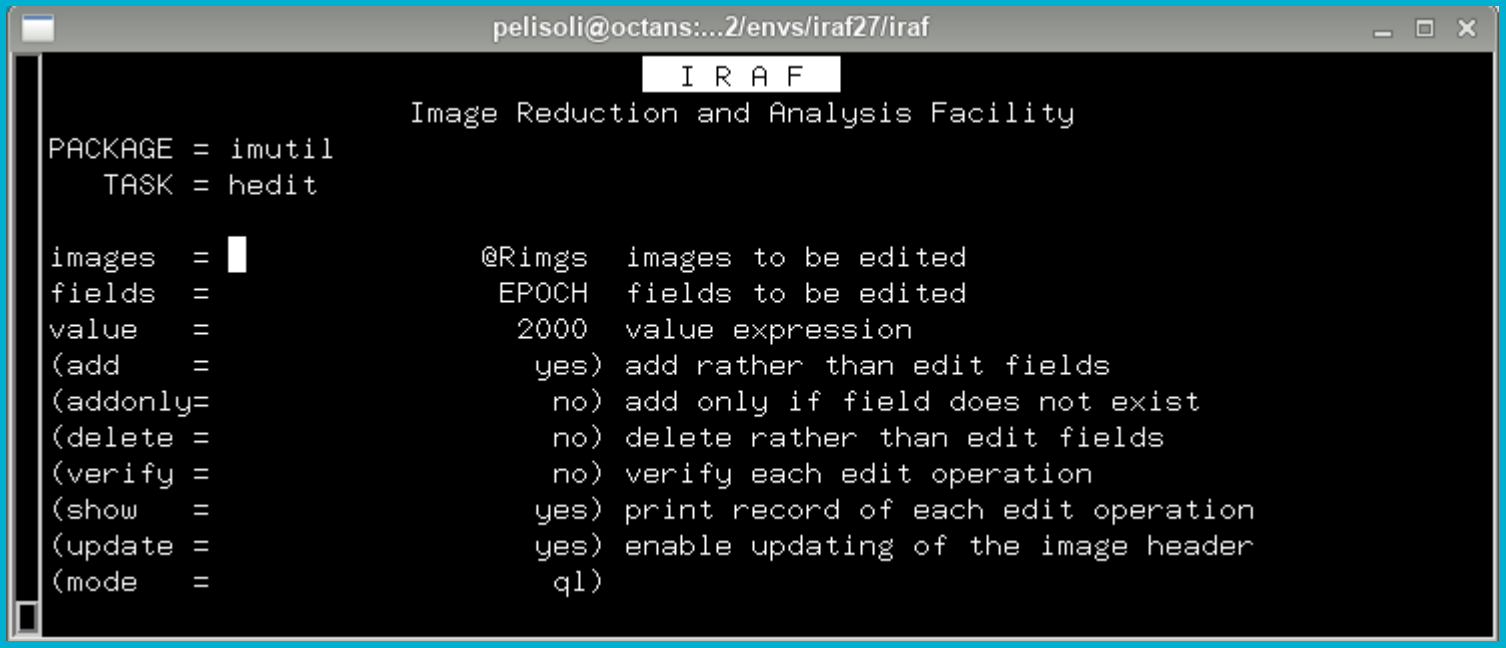
-> Chosen comparisons should be fairly constant!

-> Mean magnitudes of each star will be printed to the terminal

Building the light curve

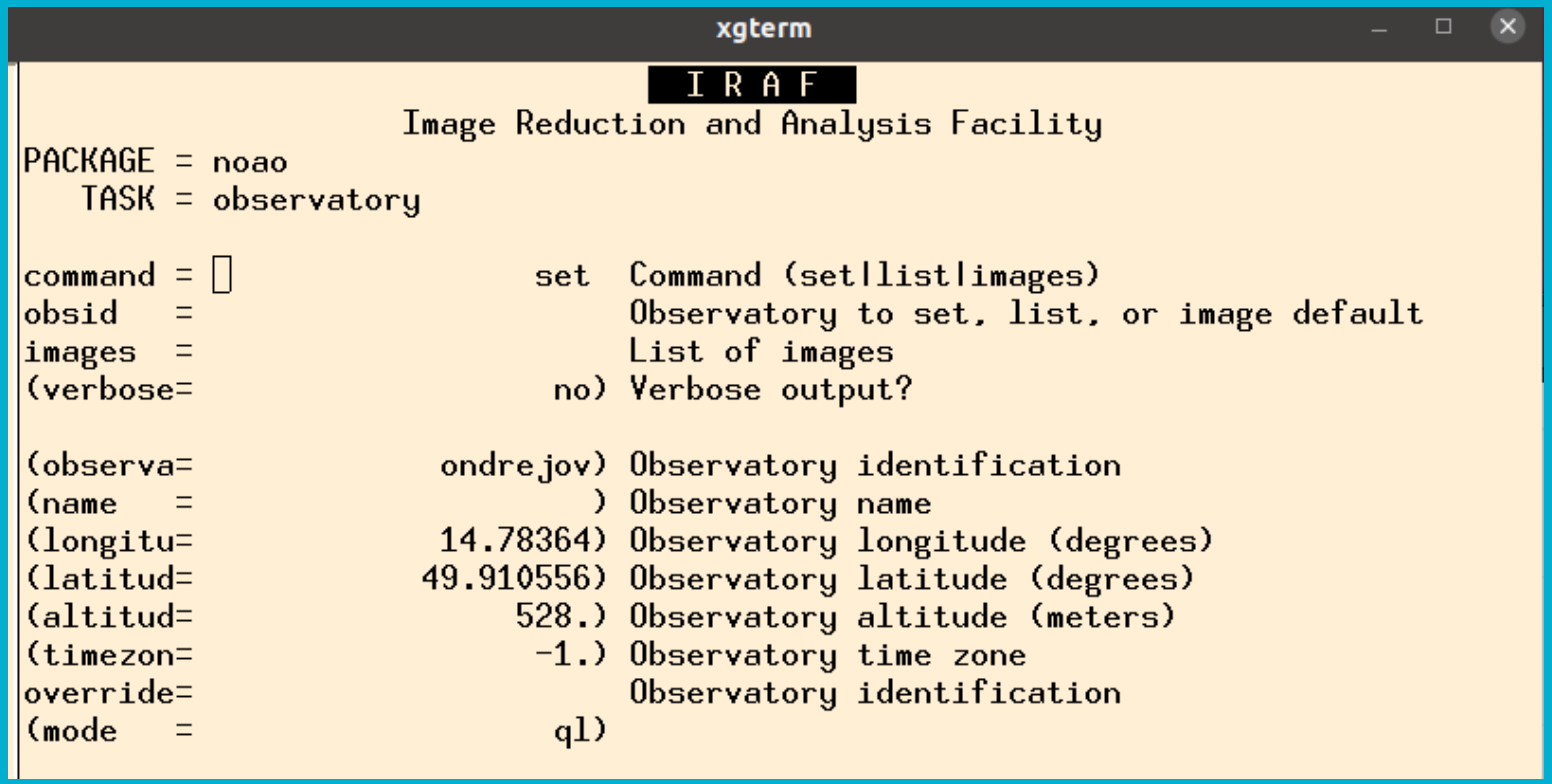
RA and DEC in a format
279.8767083356 (18h:39m:30.4s)
-5.902749998734 (-5d:54m:09.8s)

- To turn our measurements into a light curve, we need the times for each observation. We will use the task **setjd** to obtain that.
- The headers of our images are missing one important information: coordinates (RA, DEC, Epoch). Use the task **hedit** to add those to all images.



Building the light curve

- We also need to set the observatory parameters to be used for setjd. We do that with the task **observatory**:



Exit “ctrl+d” or “:go”

Building the light curve (setjd)

```
het: McDonald Observatory - Hobby-Eberly Telescope
jcdo: Jack C. Davis Observatory, Western Nevada College
lno: Langkawi National Observatory
obspars:  Use parameters from OBSERVATORY task

Observatory identification (ondrejov):
```

pelisoli@octans:...2/envs/iraf27/iraf

```
                                    I R A F
                        Image Reduction and Analysis Facility
PACKAGE = onedspec
   TASK = setjd

images  =                    @Rimgs  Images
(observa=                  obspars) Observatory of observation
(date   =                 date-obs) Date of observation keyword
(time   =                       ut) Time of observation keyword
(exposur=                  exptime) Exposure time keyword
(ra     =                       ra) Right ascension (hours) keyword
(dec    =                      dec) Declination (degrees) keyword
(epoch  =                    epoch) Epoch (years) keyword

(jd     =                       jd) Output Julian date keyword
(hjd    =                      hjd) Output Heliocentric Julian date keyword
(ljd    =                      ljd) Output local Julian date keyword

(utdate =                      yes) Is observation date UT?
(uttime =                      yes) Is observation time UT?
(listonl=                       no) List only without modifying images?
(mode   =                       ql)

                                                        ESC-? for HELP
```

Hold down the enter key until all images have been done.

setjd > R_jd (R_jd is an output file. Check it!)

Building the light curve

Check your column numbers! Might be different.

- To do differential photometry, we need to normalise the magnitudes of the star and of the comparison stars. First, check what is the average magnitude (6th column):

All the following commands have been compiled in ‘process_lightcurve.sh’!

(**5**: mag column, **6**: mag error column)

- Repeat that for all the comparison stars, and combine them into one file:

! paste mag_comp1 mag_comp2 mag_comp3 mag_comp4 > all_comp

- Average the comparison stars:

! awk '{printf "%7.4f %6.4f\n", ($1+$3+$5+$7)/4.0, sqrt($2*$2+$4*$4+ $6*$6+$8*$8)}' all_comp > mag_comp

Building the light curve

Check your column numbers! Might be different.

- To do differential photometry, we need to normalise the magnitudes of the star and of the comparison stars. First, check what is the average magnitude (6th column):

! awk '{sum+=$5;n++} END {print sum/n;}' R_star

- Then subtract it from each value: replace with calculated average

! awk '{printf "%7.4f %6.4f\n", $5-18.7529, $6}' R_star > mag_star

(**5**: mag column, **6**: mag error column)

- Repeat that for all the comparison stars, and combine them into one file:

! paste mag_comp1 mag_comp2 mag_comp3 mag_comp4 > all_comp

- Average the comparison stars:

! awk '{printf "%7.4f %6.4f\n", ($1+$3+$5+$7)/4.0, sqrt($2*$2+$4*$4+ $6*$6+$8*$8)}' all_comp > mag_comp

Building the light curve

- Combine the magnitudes of the star and the comparison magnitude:

! paste mag_star mag_comp > comb_mag

- Subtract the comparison from the star to remove background variations:

! awk '{printf "%7.4f %6.4f\n", ($1-$3), sqrt($2*$2+$4*$4)}' comb_mag > diff_mag

- Select the column containing the Heliocentric Julian Date from the file created with setjd:

awk '!/#/ {print $3}' R_jd > R_hjd

(Type this in a regular terminal. Does not work in IRAF)

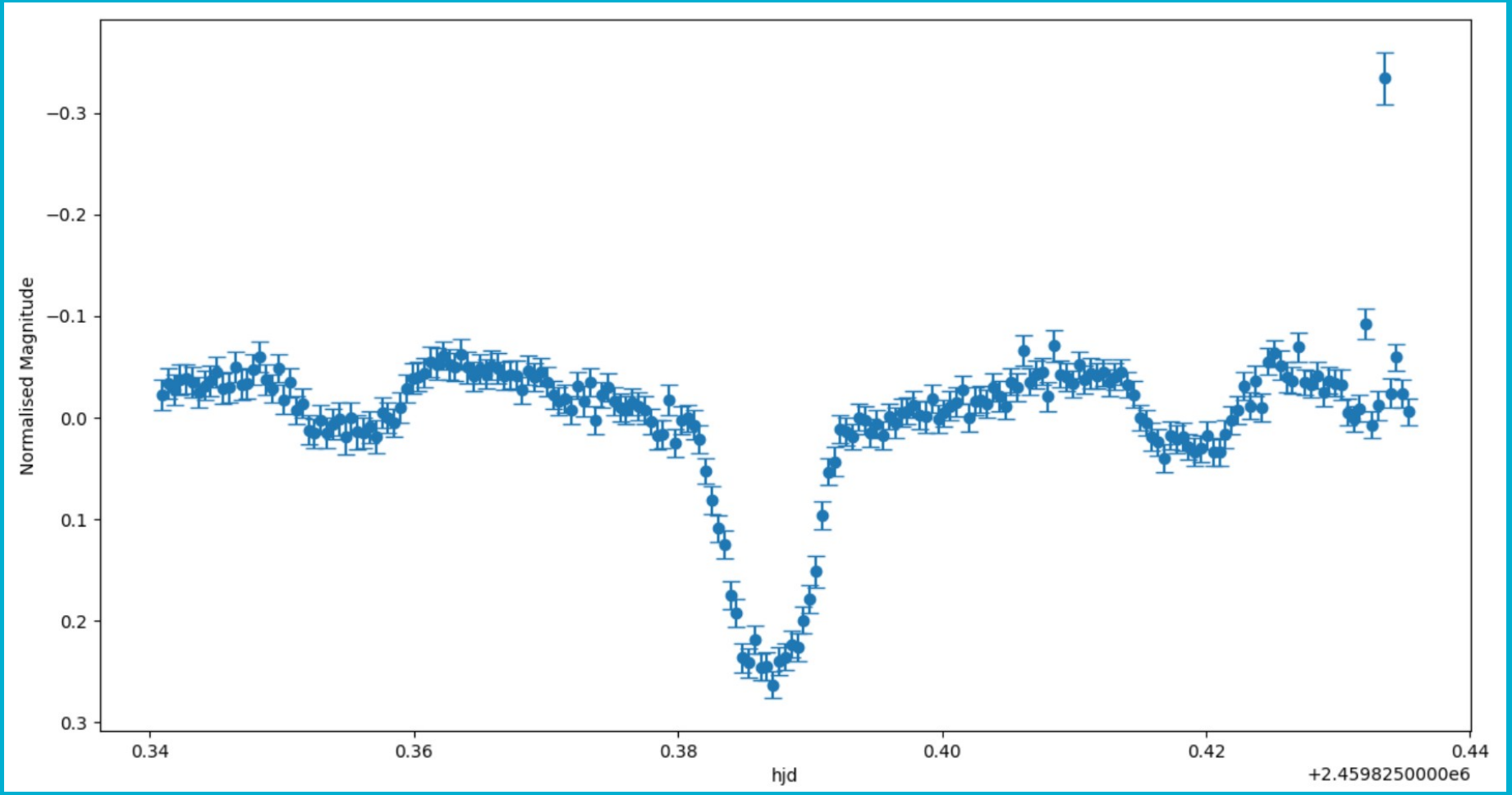
- Combine that with the magnitude to obtain the lightcurve:

! paste R_hjd diff_mag > R_lightcurve

Voilà! Now you have a light curve.

Repeat the same for the other filter.

Light curve – ta da!



Photometry – summary

- Create master files for bias, flat, and dark (zerocombine, flatcombine, darkcombine).
- Reduce the science images using ccdproc.
- Measure sky and FWHM with imexamine.
- Use the dask daofind to find the stars; do not forget to change the datapars according to your measurements, and set the threshold in findpars.
- Use the task phot to do the photometry; do not forget to update centerpars, fitskypars and photpars.
- Check ID for your star and comparison stars using display and tvmark.
- Inspect the coordinates for the star and comparison stars to guarantee there was no misidentification.
- Inspect the magnitudes of the comparison stars; they should be fairly constant.
- Use observatory and setjd to obtain the times of observation.
- Paste the times and differential magnitude (star - averaged comparison) into one file to obtain the light curve.