

Data reduction II

Photometry with IRAF

Harry Dawson

Research workshop on evolved stars
September 2025

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 σ_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 σ_i/\sqrt{n} .
- Therefore, the first step in data reduction is to calculate the average for BIAS, FLAT, and DARK images.

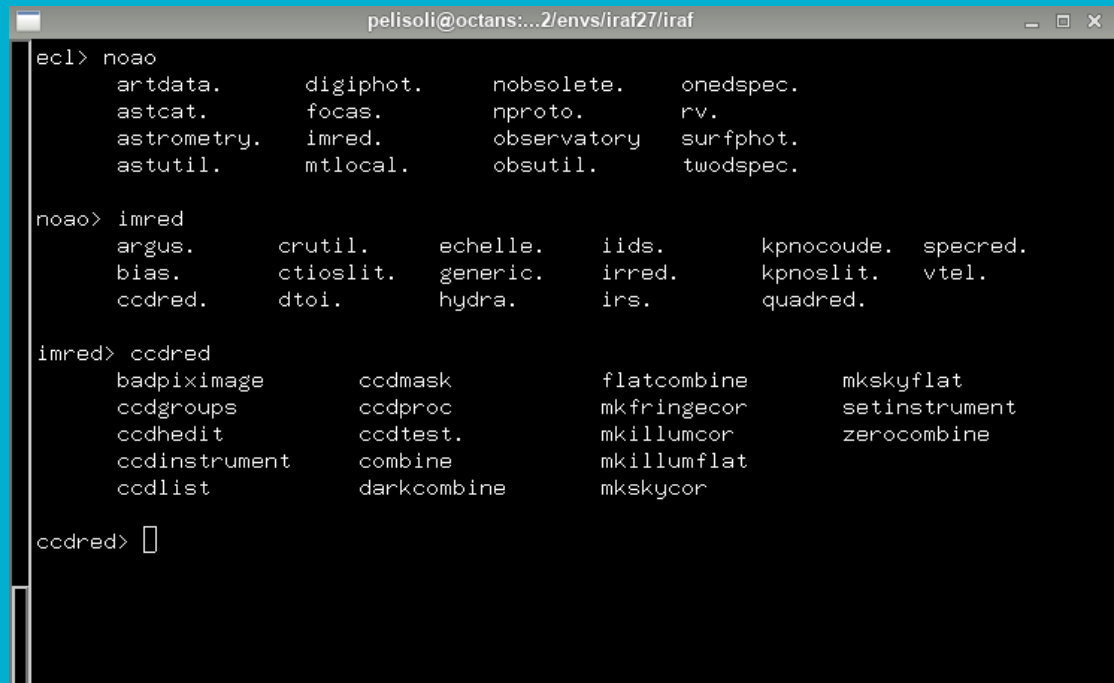
BIAS: not available for the Perek telescope.

FLAT: master flat has already been created.

DARK: we need to calculate the median dark.

IRAF – Image Reduction and Analysis Facility

- 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.



```
pelisoli@octans:...2/envs/iraf27/iraf
ec1> noao
      artdata.      digiphot.      nobsolete.      onedspec.
      astcat.       focas.         nproto.        rv.
      astrometry.   imred.        observatory    surfphot.
      astutil.      mtlocal.      obsutil.       twodspec.

noao> imred
      argus.      crutil.      echelle.      iids.      kpnocoude.  specred.
      bias.       ctioslit.    generic.      irred.     kpnoslit.   vtet.
      ccdred.     dtol.       hydra.       irs.       quadred.

imred> ccdred
      badpiximage      ccdmask      flatcombine      mkskyflat
      ccdgroups        ccdproc      mkfringecon      setinstrument
      ccdhedit         ccdtest.    mkillumcor       zerocombine
      ccdinstrument    combine     mkillumflat
      ccdlist          darkcombine mkskycon

ccdred> 
```

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

— This is what you'll get...

[illegible]

Contains science frames, darks, flats...

Check your data

— This is what you'll get...

```
!python3 sort_full.py
```

[illegible]

Contains science frames, darks, flats...

Check your data

epar display

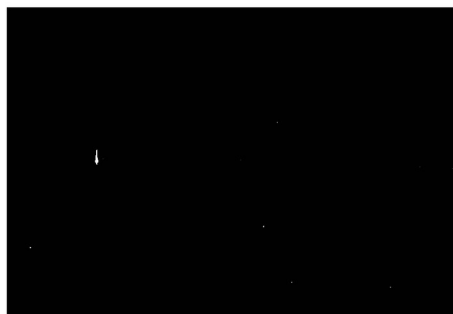


Image Reduction and Analysis Facility

```
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
(nsampl=         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 (science, darks, flats)

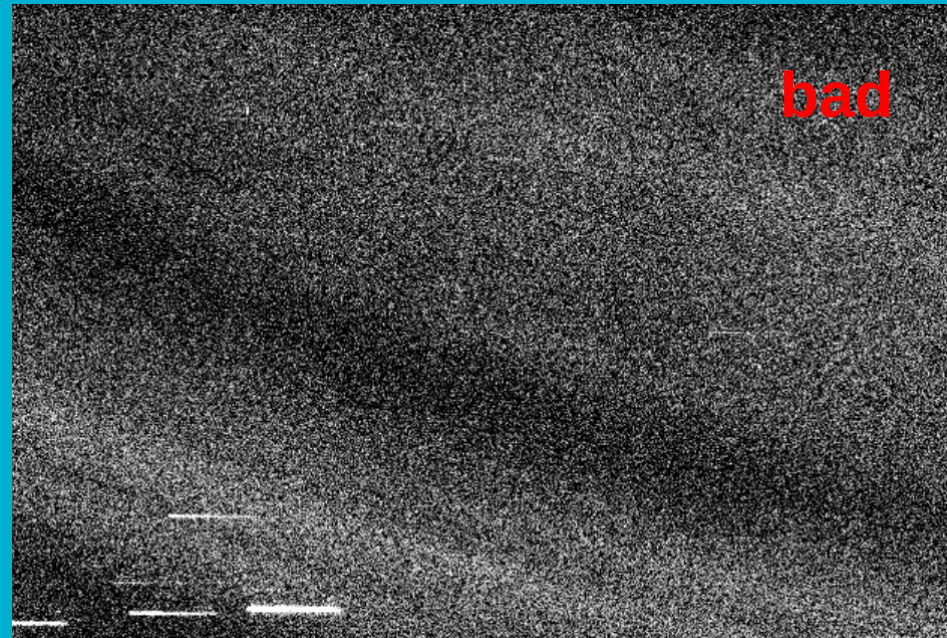
- After sorting, the data should be in different directories, check all!

!ds9 &

- Create a list with e.g. science frames (do this for darks and flats too).
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!!



Unless it goes straight
through your star!



Fixing the fits headers (optional?)

- 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 imagetype '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**
Multiple frames: **hsl Object*fits \$I,exptime yes**
("exptime" is the fits header keyword)
- Which dark images should we use?
imhead df-* lo+ | grep EXPTIME or **hsl**
- 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



Each exposure time needs its own master dark!



Creating a master dark – epar darkcombine

Check
parameters!

ccdtype has
to be empty!
Just put space.

```
pelisoli@octans:...2/envs/iraf27/iraf
IRAF
Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = darkcombine

input = 
(output = 
(combine=
(reject =
(ccdtype=
(process=
(delete =
(clobber=
(scale =
(statsec=
(nlow =
(nhigh =
(nkeep =
(mclip =
(lsigma =
(hsigma =
(rdnoise=
(gain =
(snoise =
(pclip =
(blank =
(mode =

@dark List of dark images to combine
Dark) Output dark image root name
median) Type of combine operation
sigclip) Type of rejection
) CCD image type to combine
no) Process images before combining?
no) Delete input images after combining?
no) Clobber existing output image?
exposure) Image scaling
) Image section for computing statistics
0) minmax: Number of low pixels to reject
1) minmax: Number of high pixels to reject
1) Minimum to keep (pos) or maximum to reject (neg)
yes) Use median in sigma clipping algorithms?
5.) Lower sigma clipping factor
5.) Upper sigma clipping factor
0.) ccdclip: CCD readout noise (electrons)
1.3) ccdclip: CCD gain (electrons/DN)
0.) ccdclip: Sensitivity noise (fraction)
-0.5) pclip: Percentile clipping parameter
0.) Value if there are no pixels
ql)

:go
```

“: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.
```

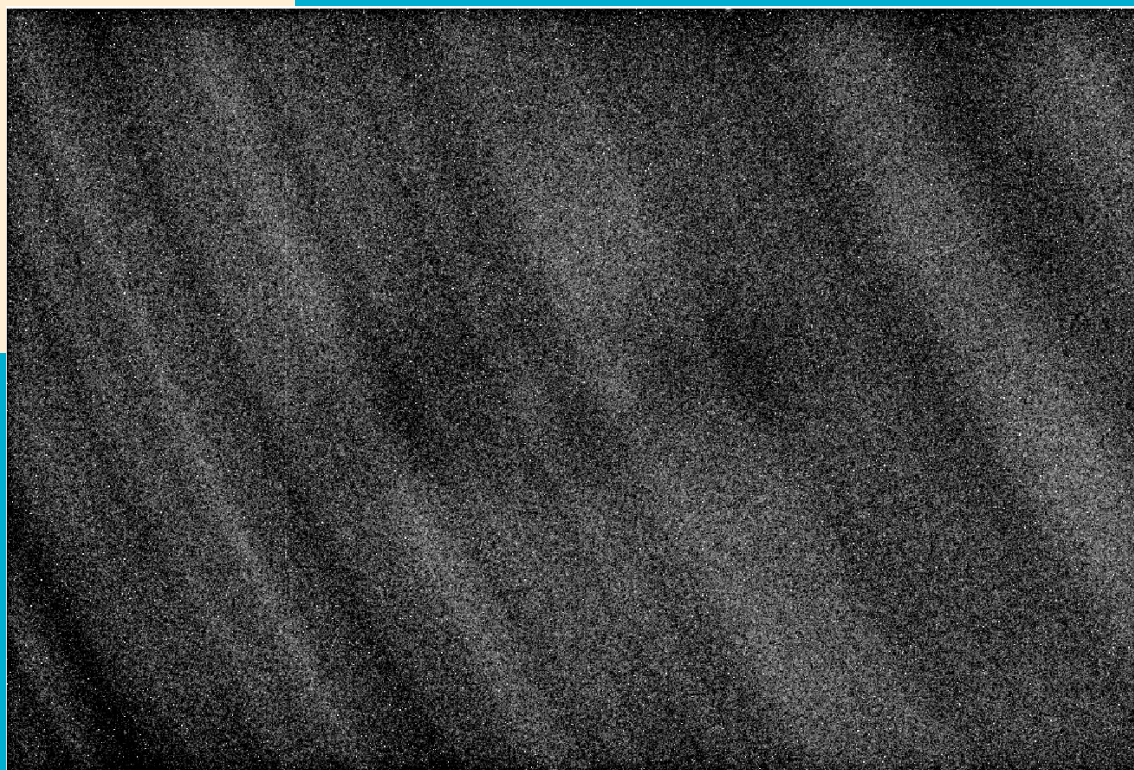
Images

```
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
```

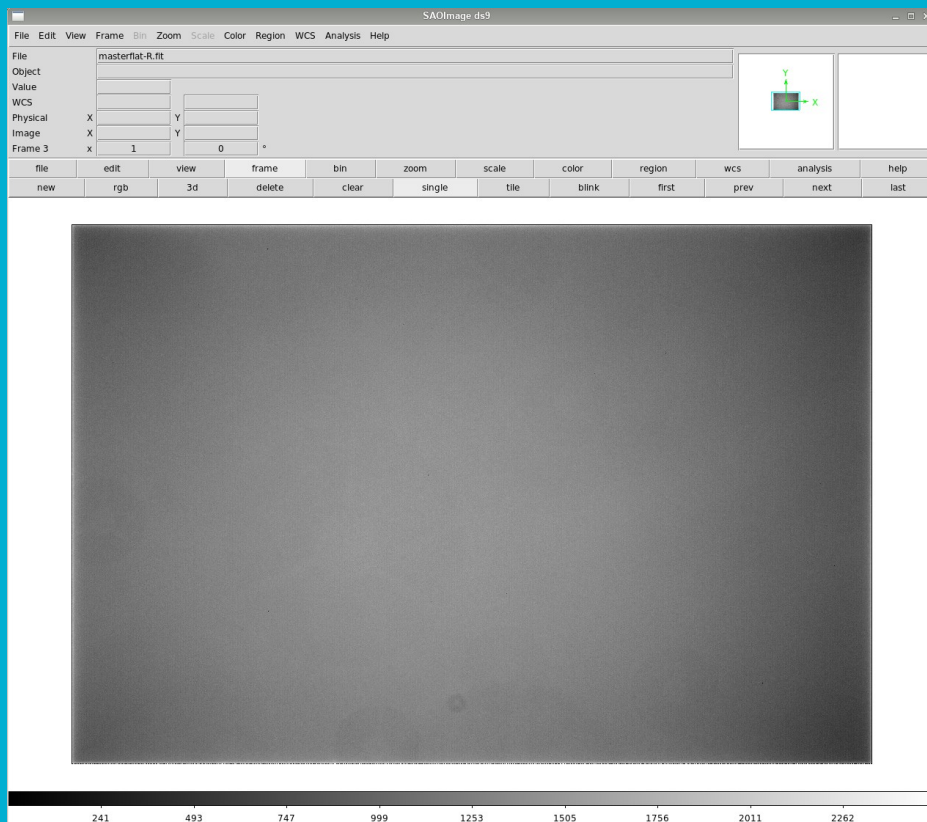
```
Output image = Dark, ncombine = 10
```

- Check what you got!

display
imstat



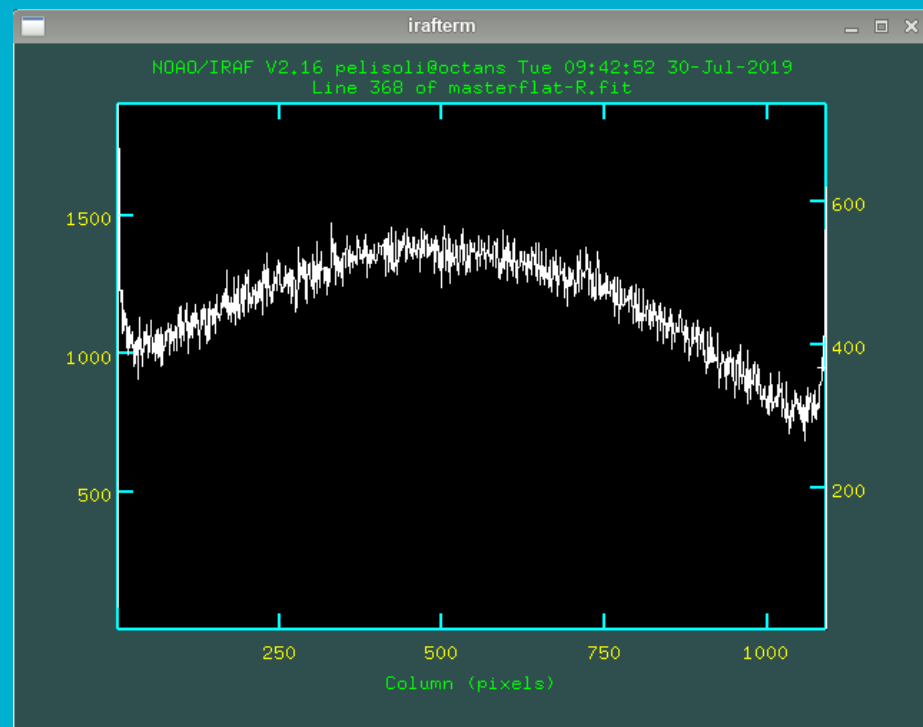
Master flat



Each filter AND exptime needs its own master flat!

Same procedure as the the master dark but use the 'flatcombine' task.

epar flatcombine



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.
(Remember to subtract your masterdark from the masterflat!)

Subtract the master dark from the master flat -> **epar ccdproc**

```
Visit http://iraf.net if you have questions or to report problems.

The following commands or packages are currently defined:

(Updated on 2013-12-13)

    adccdr.  deitab.  images.  mtools.  softtools.  upsquid.
    cfh12k.  esowfi.  kepler.  nfextern.  squid.    utilities.
    cirred.  finder.  language.  noao.    stecf.    vo.
    ctio.    fitsutil.  lists.  obsolete.  stsdas.  xdimsum.
    cutoutpkg.  gemini.  mem0.   plot.    system.  xray.
    dataio.  gmisc.  mscdb.  proto.   tables.
    dbms.    guiapps.  mscred.  rvsao.   ucscrir.

ecl> ls
bin      isis_grids  muniwin  Share      uparm
clean    isis_installation  nanorcs  snap
Desktop  kinematics  pfiles   spas
                                I R A F
                                Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = ccdproc

images =          Flat.fit  List of CCD images to correct
(output =         cFlat.fit) List of output CCD images
(ccdtype=         ) CCD image type to correct
(max_cac=         0) Maximum image caching memory (in Mbytes)
(noproc =         no) List processing steps only?

(fixpix =         no) Fix bad CCD lines and columns?
(oversca=        no) Apply overscan strip correction?
(trim =          no) Trim the image?
(zerozor=        no) Apply zero level correction?
(darkcor=        yes) Apply dark count correction?
(flatcor=        no) Apply flat field correction?
(illumco=        no) Apply illumination correction?
(fringec=        no) Apply fringe correction?
(readcor=        no) Convert zero level image to readout correction?
(scancor=        no) Convert flat field image to scan correction?

(readaxi=         line) Read out axis (column\line)
(fixfile=         ) File describing the bad lines and columns
(biassec=         ) Overscan strip image section
(trimsec=         ) Trim data section
(zero =           ) Zero level calibration image
(dark =           ../Dark_60s.fit) Dark count calibration image
(flat =           ) Flat field images
(illum =          ) Illumination correction images
(fringe =         ) Fringe correction images

More
```

Continues ...

Reducing the science images 1 - epar ccdproc

Check
parameters!

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = ccdproc

images = [] @Vings List of CCD images to correct
(output = c//@Vings) List of output CCD images
(ccdtype= ) CCD image type to correct
(max_cac= 0) Maximum image caching memory (in Mbytes)
(noproc = no) List processing steps only?

(fixpix = no) Fix bad CCD lines and columns?
(oversca= no) Apply overscan strip correction?
(trim = no) Trim the image?
(zeroeor= no) Apply zero level correction?
(darkcor= yes) Apply dark count correction?
(flatcor= yes) Apply flat field correction?
(illumco= no) Apply illumination correction?
(fringec= no) Apply fringe correction?
(readcor= no) Convert zero level image to readout correction?
(scancor= no) Convert flat field image to scan correction?

(readaxi= line) Read out axis (column|line)
(fixfile= ) File describing the bad lines and columns
(biassec= ) Overscan strip image section
(trimsec= ) Trim data section

More
ESC-? for HELP
```

Continues ...

Reducing the science images 2 - epar ccdproc

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = ccdred
TASK = ccdproc
More
(zero = ) Zero level calibration image
(dark =  Dark.fits) Dark count calibration image
(flat =  masterflat-V.fit) Flat field images
(illum = ) Illumination correction images
(fringe = ) Fringe correction images
(minrepl=  1.) Minimum flat field value
(scantyp=  shortscan) Scan type (shortscan|longscan)
(nscan =  1) Number of short scan lines

(interac=  no) Fit overscan interactively?
(function=  legendre) Fitting function
(order =  1) Number of polynomial terms or spline pieces
(sample =  *) Sample points to fit
(naverag=  1) Number of sample points to combine
(niterat=  1) Number of rejection iterations
(low_rej=  3.) Low sigma rejection factor
(high_re=  3.) High sigma rejection factor
(grow =  0.) Rejection growing radius
(mode =  ql)

:go -> reduced files will start with a 'c'
```

ESC-? for HELP

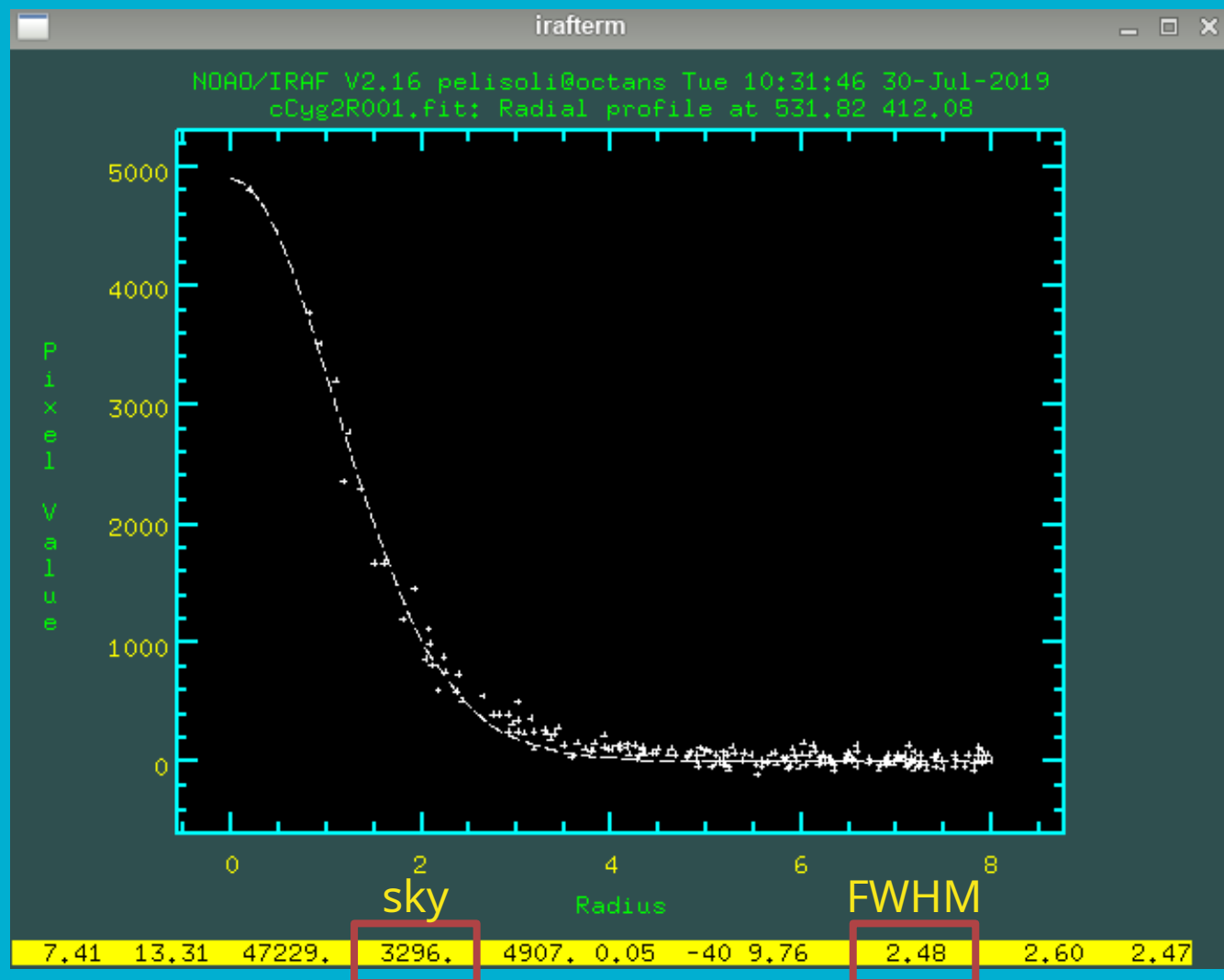
Photometry

- 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



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)

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = daofind

image = cCyg2R001.fit,cCyg2R223.fit Input image(s)
output = default Output coordinate file(s) (default: image.coo.?)
(starmap= ) Output density enhancement image(s)
(skymap = ) Output sky image(s)
(datapar=:e ) Data dependent parameters
(findpar=:e ) Object detection parameters
(boundar= nearest) Boundary extension (constant|nearest|reflect|wra
(constan= 0.) Constant for boundary extension
(interac= no) Interactive mode?
(icomman= ) Image cursor: [x y wcs] key [cmd]
(gcomman= ) Graphics cursor: [x y wcs] key [cmd]
(wcsout = )_.wcsout) The output coordinate system (logical,tv,physica
(cache = )_.cache) Cache the image pixels?
(verify = )_.verify) Verify critical daofind parameters?
(update = )_.update) Update critical daofind parameters?
(verbose= )_.verbose) Print daofind messages?
(graphic= )_.graphics) Graphics device
(display= )_.display) Display device
(mode = ql)

ESC-? for HELP
```

DAOFIND (datapars)

“:q” to go back

```
pelisoli@octans:...2/envs/iraf27/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = datapars

(scale = 1.) Image scale in units per pixel
(fwhmpsf= 2.5) FWHM of the PSF in scale units
(emissio= yes) Features are positive?
(sigma = 20.) Standard deviation of background in counts
(datamin= INDEF) Minimum good data value
(datamax= INDEF) Maximum good data value
(noise = poisson) Noise model
(ccdread= ) CCD readout noise image header keyword
(gain = GAIN) CCD gain image header keyword
(readnoi= 0.) CCD readout noise in electrons
(epadu = 1.3) Gain in electrons per count
(exposur= EXPTIME) Exposure time image header keyword
(airmass= ) Airmass image header keyword
(filter = FILTER) Filter image header keyword
(obstime= UT) Time of observation image header keyword
(itime = 1.) Exposure time
(xairmas= INDEF) Airmass
(ifilter= INDEF) Filter
(otime = INDEF) Time of observation
(mode = ql)

ESC-? for HELP
```

DAOFIND (findpars)

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = findpars
(thresho= 5.) Threshold in sigma for feature detection
(nsigma = 1.5) Width of convolution kernel in sigma
(ratio = 1.) Ratio of minor to major axis of Gaussian kernel
(theta = 0.) Position angle of major axis of Gaussian kernel
(sharplo= 0.2) Lower bound on sharpness for feature detection
(sharphi= 1.) Upper bound on sharpness for feature detection
(roundlo= -1.) Lower bound on roundness for feature detection
(roundhi= 1.) Upper bound on roundness for feature detection
(mkdetec= no) Mark detections on the image display?
(mode = ql)
```

How many sigma above the background.
Usually a value between 4 and 6 is good for
finding all stars. To restrict to brighter stars,
use a larger value.

ESC-? for HELP

DAOFIND (findpars)

```
pelisoli@octans:...2/envs/iraf27/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = findpars

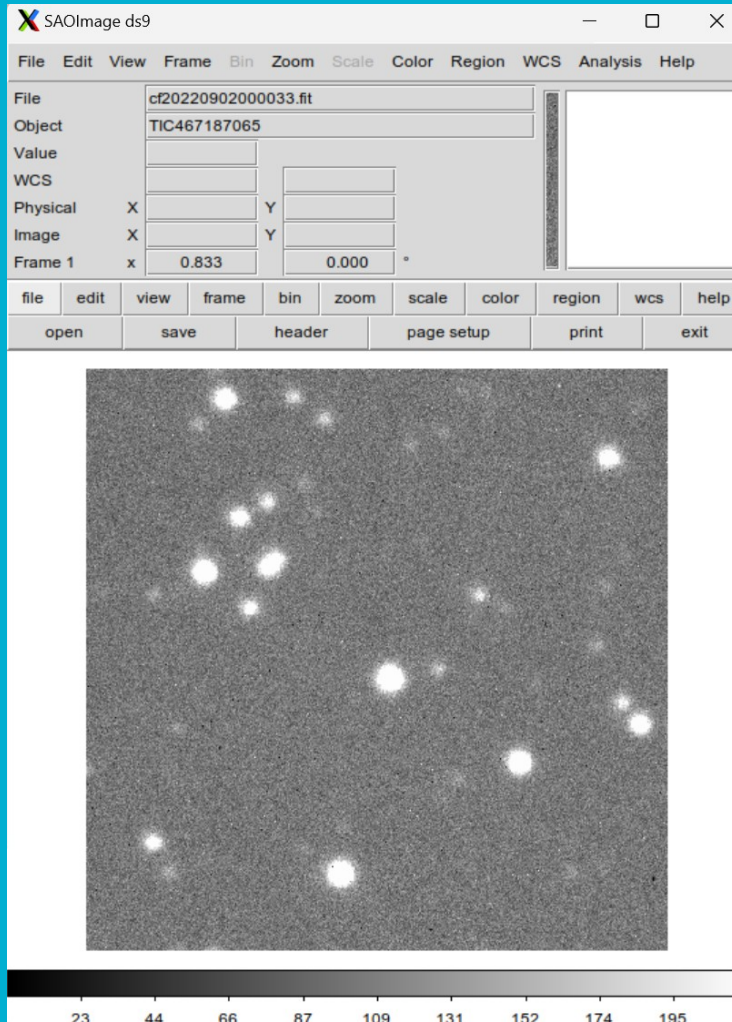
(thresho=      5.) Threshold in sigma for feature detection
(nsigma =      1.5) Width of convolution kernel in sigma
(ratio  =      1.) Ratio of minor to major axis of Gaussian kernel
(theta  =      0.) Position angle of major axis of Gaussian kernel
(sharplo=    0.2) Lower bound on sharpness for feature detection
(sharphi=      1.) Upper bound on sharpness for feature detection
(roundlo=    -1.) Lower bound on roundness for feature detection
(roundhi=      1.) Upper bound on roundness for feature detection
(mkdetec=    no) Mark detections on the image display?
(mode   =      ql)

Output: *fits.coo.1

tdump cCyg2R001.fit.coo.1 columns=c1,c2,c7 > coordsR
(I'd recommend writing these commands in a .txt file as
you'll need to go through multiple times!)
ESC-? for HELP
```

DAOFIND (tvmark: to display the found stars)

display cObject...



epar tvmark
:go

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility

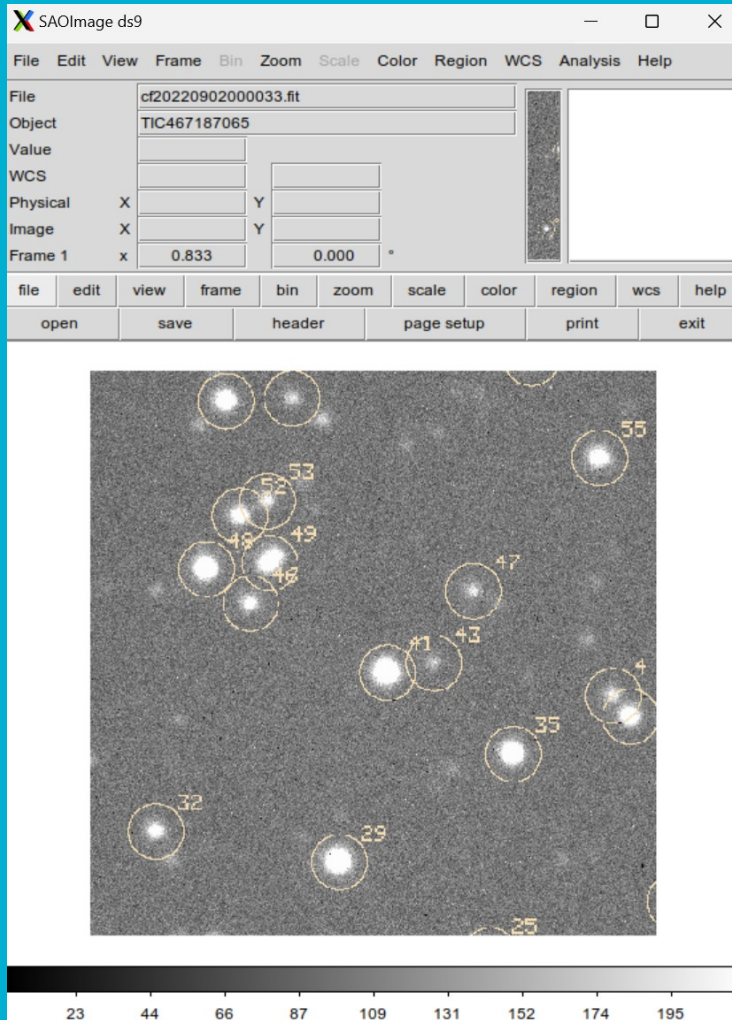
PACKAGE = tv
TASK = tvmark

frame = 1 Default frame number for display
coordsR = input coordinate list
(logfile= ) Output log file
(autolog= no) Automatically log each marking command
(outimage= ) Output snapped image
(deletio= ) Output coordinate deletions list
(command= ) Image cursor: [x y wcs] key [cmd]
(mark = circle) The mark type
(radii = 25) Radii in image pixels of concentric circles
(lengths= 0) Lengths and width in image pixels of concentric
raster) Default font
(color = 0) Gray level of marks to be drawn
(label = yes) Label the marked coordinates
(number = no) Number the marked coordinates
(nxoffse= 0) X offset in display pixels of number
(nyoffse= 0) Y offset in display pixels of number
(pointsi= 3) Size of mark type point in display pixels
(txsize = 2) Size of text and numbers in font units
(toleran= 1.5) Tolerance for deleting coordinates in image pixe
(interac= no) Mode of use
(mode = ql)

ccdred>
```

DAOFIND (tvmark: to display the found stars)

display cObject...



epar tvmark
:go

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility

PACKAGE = tv
TASK = tvmark

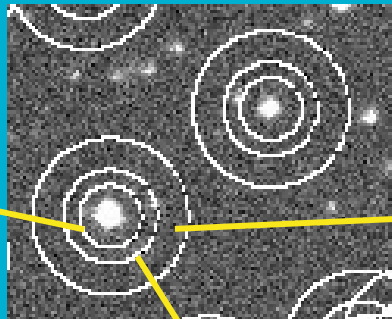
frame = 1 Default frame number for display
coords = coordsR input coordinate list
(logfile= ) Output log file
(autolog= no) Automatically log each marking command
(outimage= ) Output snapped image
(deletio= ) Output coordinate deletions list
(command= ) Image cursor: [x y wcs] key [cmd]
(mark = circle) The mark type
(radii = 25) Radii in image pixels of concentric circles
(lengths= 0) Lengths and width in image pixels of concentric
raster) Default font
(color = 0) Gray level of marks to be drawn
(label = yes) Label the marked coordinates
(number = no) Number the marked coordinates
(nxoffse= 0) X offset in display pixels of number
(nyoffse= 0) Y offset in display pixels of number
(pointsi= 3) Size of mark type point in display pixels
(txsize = 2) Size of text and numbers in font units
(toleran= 1.5) Tolerance for deleting coordinates in image pixe
(interac= no) Mode of use
(mode = ql)

ccdred> 
```

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 $\sim 2.5 \times \text{FWHM}$



Dannulus: width of the ring to count the background.
 $\sim 5\text{-}10$ pixels

Annulus (radii): distance at which to start counting the background.
At least $2.5 \times \text{FWHM}$
 $\sim 4 \times \text{FWHM}$ in our example

* For a Gaussian distribution:
 $\text{FWHM} = 2.35\sigma$
99.99% of the light is contained within
 $4\sigma = 1.7\text{FWHM}$

DAOPHOT - epar phot

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

NB!

One coordinate
file for all frames

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility

PACKAGE = daophot
TASK = phot

Image = @cimgsR_beg Input image(s)
coords = c0yg2R001.fit.coo.1 Input coordinate list(s) (default: image.coo.?)
output = default Output photometry file(s) (default: image.mag.?)
skyfile = Input sky value file(s)
(plotfil= ) Output plot metacode file
(datapar= ) Data dependent parameters
(centerp= :e ) Centering parameters
(fitskyp= :e ) Sky fitting parameters
(photpar= :e ) Photometry parameters
(interac= no) Interactive mode?
(radplot= no) Plot the radial profiles?
(icomman= ) Image cursor: [x y wcs] key [cmd]
(gcomman= ) Graphics cursor: [x y wcs] key [cmd]
(wcsin = )_.wcsin) The input coordinate system (logical,tv,physical)
(wcsout = )_.wcsout) The output coordinate system (logical,tv,physical)
(cache = )_.cache) Cache the input image pixels in memory?
(verify = )_.verify) Verify critical phot parameters?
(update = )_.update) Update critical phot parameters?
(verbose= )_.verbose) Print phot messages?
(graphic= )_.graphics) Graphics device
(display= )_.display) Display device

More

ESC-? for HELP
```


DAOPHOT (centerpars)

```
pelisoli@octans:...2/envs/iraf27/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = centerpars

(calgori=  centroid) Centering algorithm
(cbox    =      5.) Centering box width in scale units
(cthresh=      0.) Centering threshold in sigma above background
(minsnra=      1.) Minimum signal-to-noise ratio for centering algo
(cmaxite= 10) Maximum iterations for centering algorithm
(maxshif= 1.) Maximum center shift in scale units
(clean   = no) Symmetry clean before centering
(rclean  = 1.) Cleaning radius in scale units
(rclip   = 2.) Clipping radius in scale units
(kclean  = 3.) K-sigma rejection criterion in skysigma
(mkcente= no) Mark the computed center
(mode    = ql)

ESC-? for HELP
```

PHOT (fitskypars)

**Your data
specific!**

```
pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = fitskypars

(salgori= mode) Sky fitting algorithm
(annulus= 15.) Inner radius of sky annulus in scale units
(dannulu= 10.) Width of sky annulus in scale units
(skyvalu= 415.) User sky value
(smaxite= 20) Maximum number of sky fitting iterations
(sloclip= 0.) Lower clipping factor in percent
(shiclip= 0.) Upper clipping factor in percent
(snrejec= 50) Maximum number of sky fitting rejection iteratio
(sloreje= 3.) Lower K-sigma rejection limit in sky sigma
(shireje= 3.) Upper K-sigma rejection limit in sky sigma
(khist = 3.) Half width of histogram in sky sigma
(binsize= 0.1) Binsize of histogram in sky sigma
(smooth = no) Boxcar smooth the histogram
(rgrow = 0.) Region growing radius in scale units
(mksky = no) Mark sky annuli on the display
(mode = ql)

ESC-? for HELP
```

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
(apertur= 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

IMPORTANT COLUMNS

c4 = star ID

c5 = x coordinate

c6 = y coordinate

c29 = magnitude

c30 = magnitude error

c28= flux.

#	#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	##	errors
#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	##	errors	
#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	##	errors
#F	%-12.2f	%-14.7g	%-11.7g	%-14.7g	%-7.3f	%-6.3f	%-5d	%-9s
#								
	c0bject_3__R_0001.fits	873.889	2.343	1	c0bject_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
	c0bject_3__R_0001.fits	479.859	4.346	2	c0bject_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		

Make a list of photometry files:

ls *R*mag.1 > Rmag_files

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

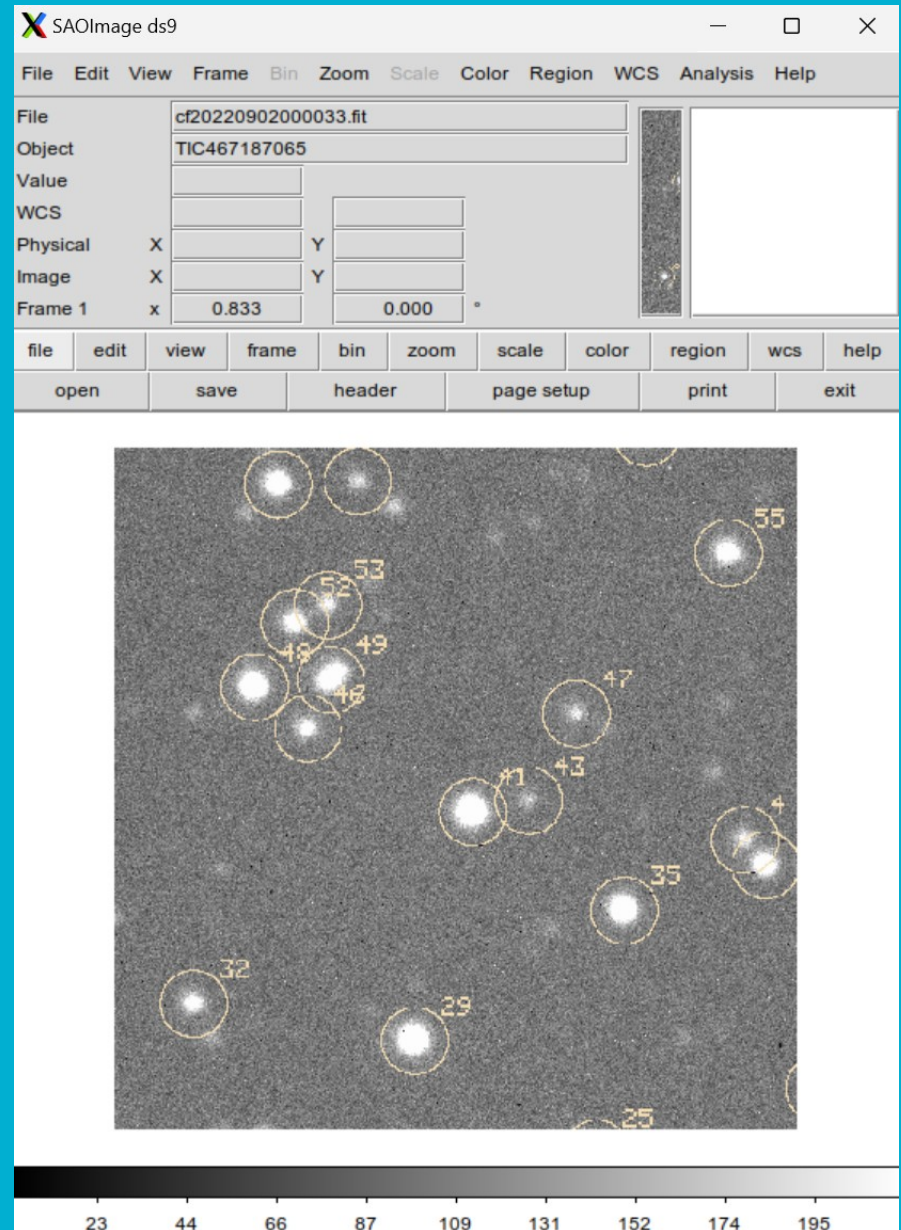
Photometry

- 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)

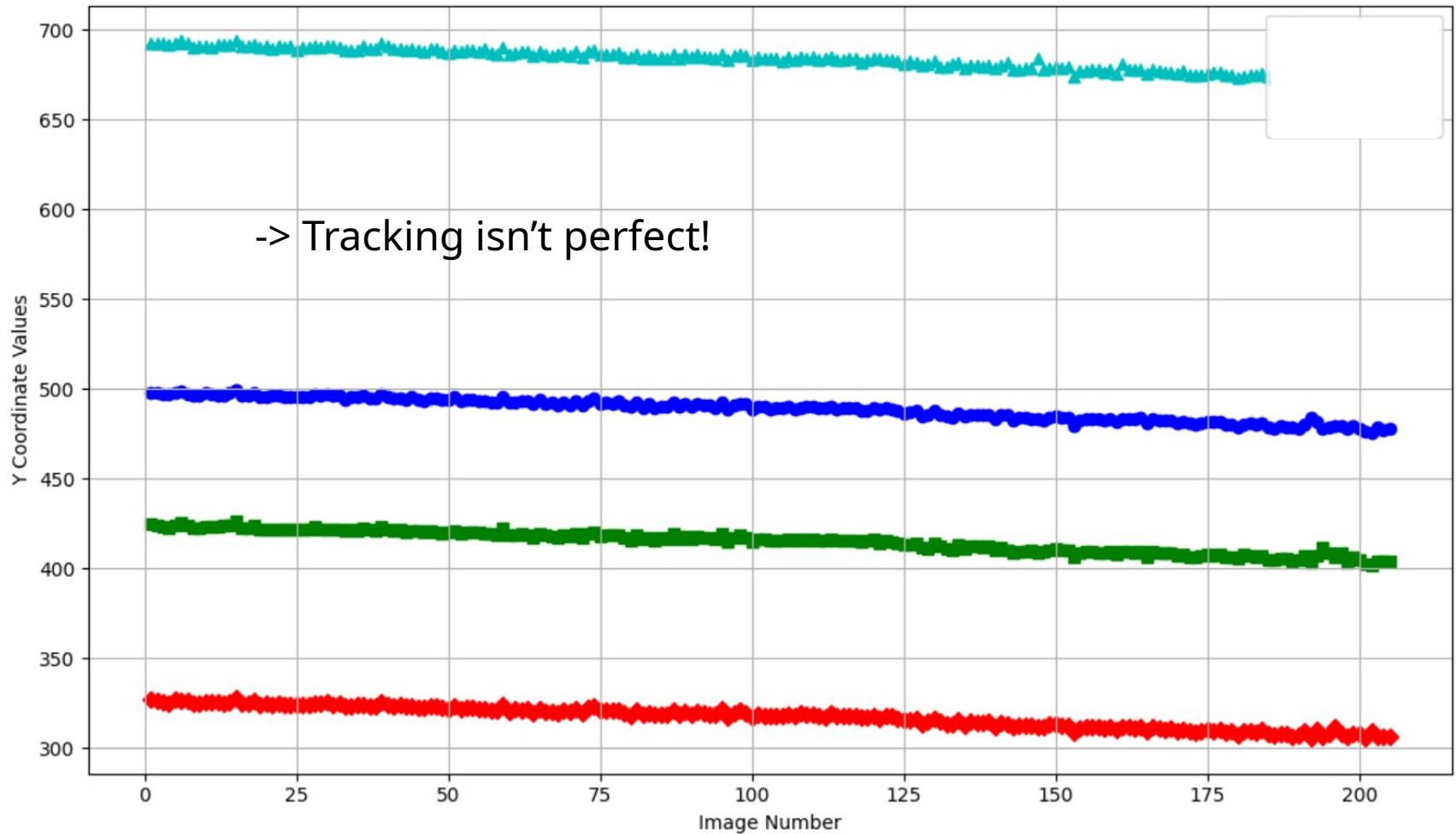
- Copy the photometry of the star and each comparison into separate files.

```
! 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
```

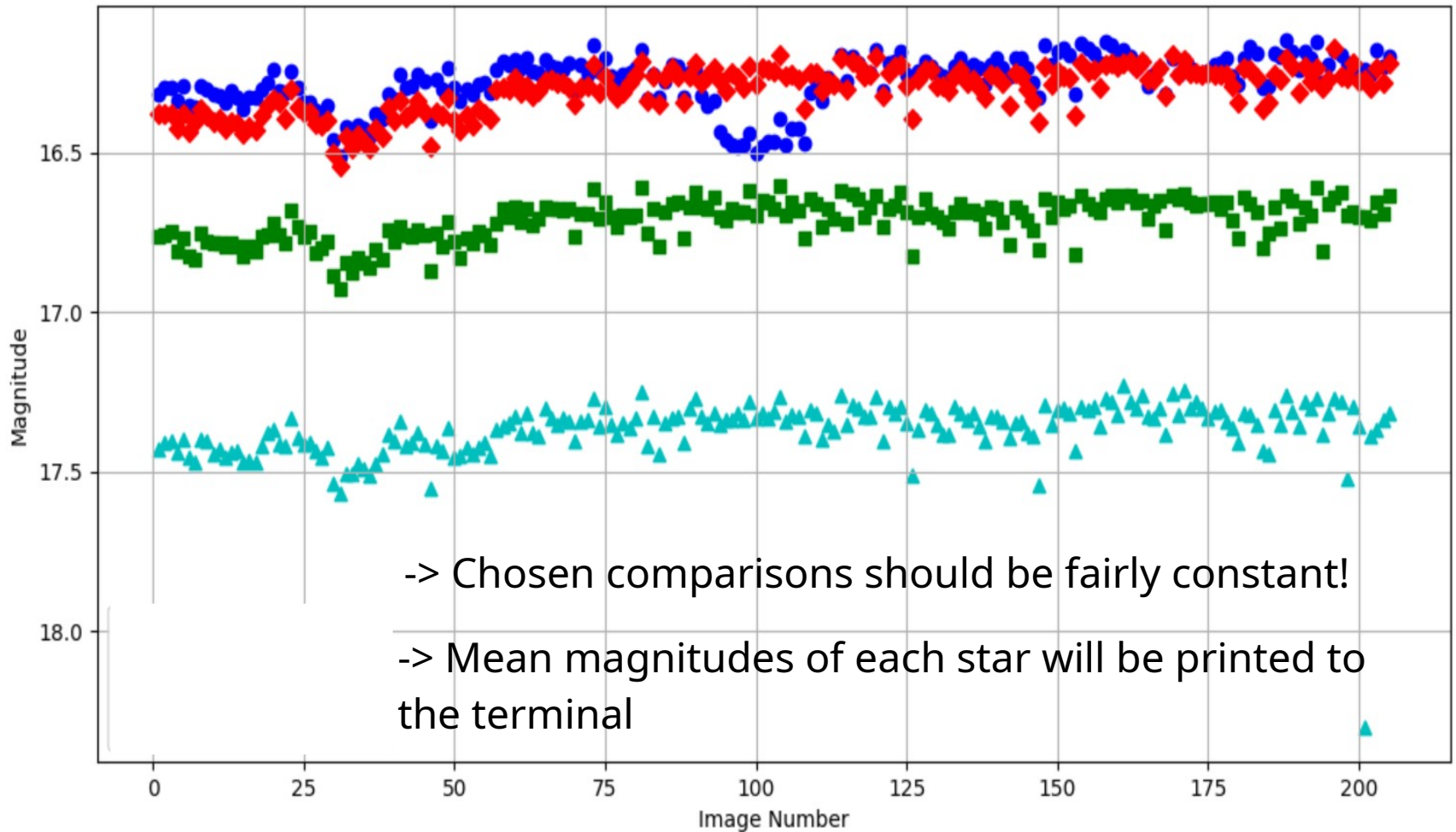
(Replace with correct Ids)

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

Photometry – shifting images



Photometry - magnitudes



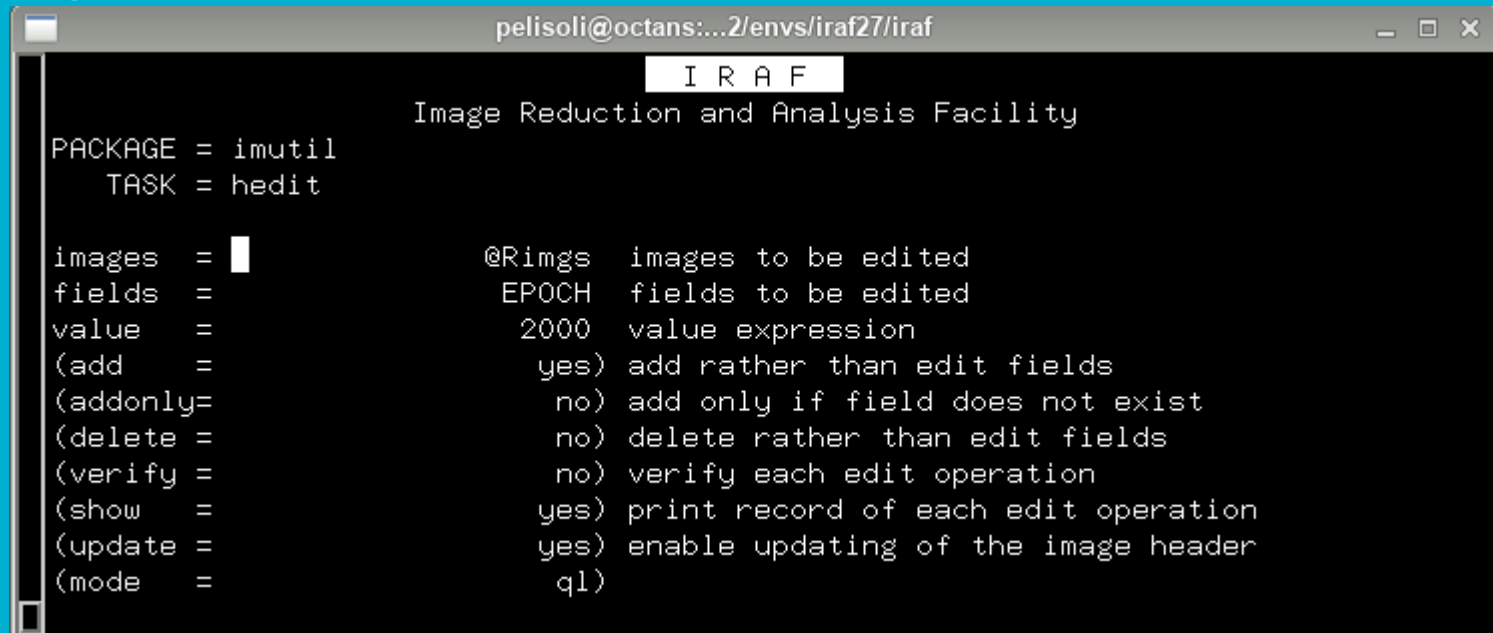
Building the light curve

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.



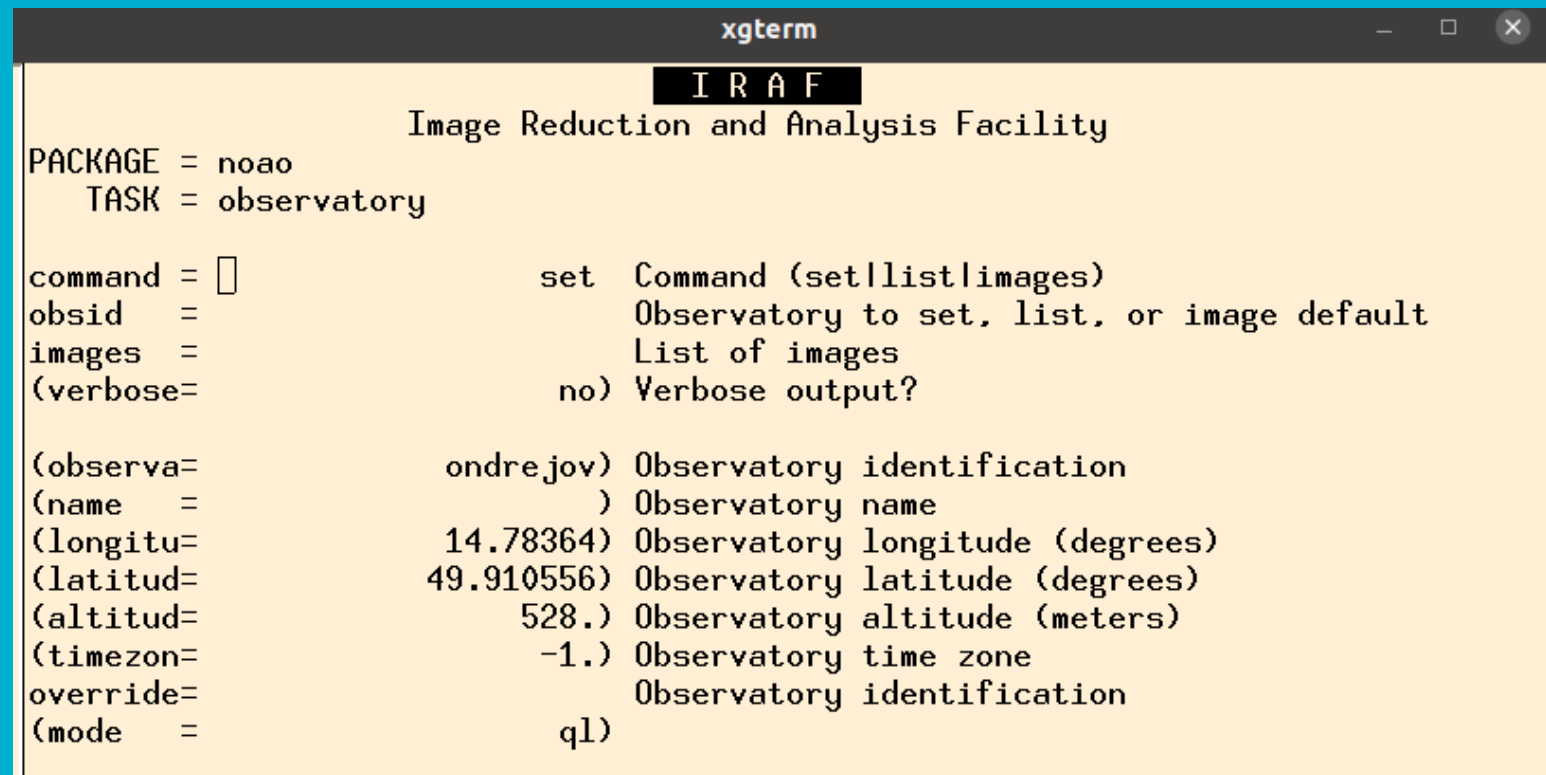
```
pelisoli@octans:...2/envs/iraf27/iraf
IRAF
Image Reduction and Analysis Facility

PACKAGE = imutil
TASK = hedit

images = 
fields = 
value = 2000
(add = yes) add rather than edit fields
(addonly= no) add only if field does not exist
(delete = no) delete rather than edit fields
(verify = no) verify each edit operation
(show = yes) print record of each edit operation
(update = yes) enable updating of the image header
(mode = ql)
```

Building the light curve

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



```
xgterm
I R A F
Image Reduction and Analysis Facility
PACKAGE = noao
TASK = observatory

command = 
obsid = set Command (setl1listimages)
          Observatory to set, list, or image default
images = List of images
(verbose= no) Verbose output?

(observa= ondrejov) Observatory identification
(name = ) Observatory name
(longitu= 14.78364) Observatory longitude (degrees)
(latitud= 49.910556) Observatory latitude (degrees)
(altitud= 528.) Observatory altitude (meters)
(timezon= -1.) Observatory time zone
override= Observatory identification
(mode = ql)
```

Exit “ctrl+d” or “:go”

Building the light curve (**setjd**)

```
het: McDonald Observatory - Hobby-Eberly Telescope
jco: Jack C. Davis Observatory, Western Nevada College
lno: Langkawi National Observatory
obsvars: 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 =
  @Rings Images
(observa=  obsvars) 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?
(listonly= 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
```

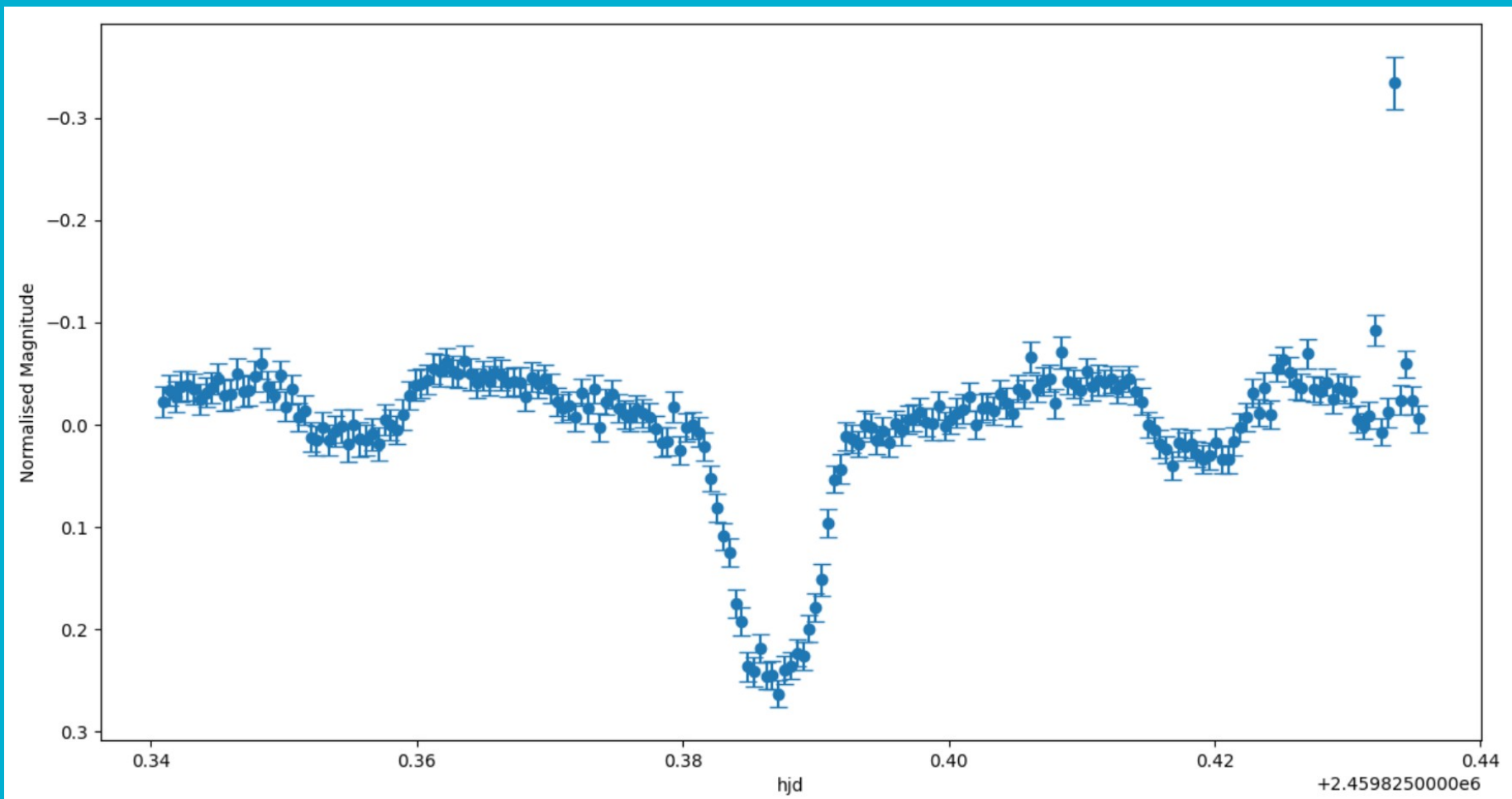
Building the light curve

- 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!



To be continued in the light curve analysis starting Friday (photometry group only)

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 task **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.