****

* Argos Data Reduction Instructions *

****

Rewritten by [redacted]
UTEP Summer 2012
From argos_data_reduction

Needed

- Installation of IRAF (to put in other computer, check *IRAF Install Guide*)
- Directory for login.cl
- Installation of login.cl
- .fits Images

For beginners in linux, check out *Quick Guide Linux and IRAF Terminal Commands.*

Open up two terminals. One terminal will start the IRAF program and the other into the folder which has your data.

Terminal icon either looks like this at the top menu bar of the screen:
or, you can do a search for terminal, xgterm, or xterm.

Check For 'login.cl'

In the terminal that your data will shown, do this procedure.

login.cl is the command to start iraf.

To check for it:

- Go into (or **cd**) iraf directory.
- List all files in directory and check to see if **login.cl** is there.

If not there, but IRAF is installed:

- Create iraf command – **mkiraf**
- Enter **xgterm**
- The **login.cl** should have been created, please check just in-case

Start IRAF and Install Packages

NOTE: It is better to start IRAF on an xgterm. On the MountainLion (iMac), the X11 needs to be installed. After that install the xgterm and ds9. On the normal terminal, enter:

<username>@...$ xgterm

This will open a xgterm.

In the terminal (xgterm or xterm) to start the IRAF program.

Enter into IRAF folder and open:

<username>@...$ cd ~/iraf //this is an example of an IRAF folder
<username>@...$ cl //will open iraf
<username>@...$ pkgname1 pkgname2 ...etc

ecl>

Enter:
noao
imred
ccdred
digiphot
apphot

Line will show: **apphot >**

In the other terminal:

cd /home or **/home/<username>/Desktop/McDonald** //if data is on Desktop

cd /home/McDonald/mcdred/<date> //if data is in **home** folder/directory

Begin by creating lists of files to be processes. In the following it is assumed that you have several **Dark** exposure times and **flat** filter runs and the source files have names starting with star you are working with. Ex. J1753 or J1118. EDIT THE FOLLOWING ACCORDING TO YOUR FILES THEN COPY AND PASTE.

ls b*
ls d*
ls f*
ls J1118*.fits //example for star J1118
ls J1753*.fits //example for star J1753

If screen shows:
ls: cannot access b*: No such file or directory //try to see if runs are capitalized

ls bias*.fits > biaslist or **ls Bias*.fits > biaslist**

There will be different darks so list all darks to see which ones exist
ls dark.*.fits > darklist or **ls Dark*.fits > ...etc**
ls dark1.*.fits > dark1list
ls dark2.*.fits > dark2list
etc.

ls flat.*.fits > flatlist or **ls flat1.*.fits > flatlist**

ls J1118*.fits > J1118list

Make sure that the following settings are in place.

apphot > epar zerocombine

I R A F
Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = zerocombine

input = @biaslist List of zero level images to combine
(output = zero) Output zero level name
(combine = median) Type of combine operation
(reject = minmax) Type of rejection
(ccdtype = none) CCD image type to combine
(process = no) Process images before combining?
(delete = no) Delete input images after combining?
(clobber = no) Clobber existing output image?
(scale = none) Image scaling
(statsec =) Image section for computing statistics
(nlow = 0) minmax: Number of lows pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise = 0.) ccdclip: CCD readout noise (electrons)
(gain = 1.) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 0.) Value if there are no pixels
(mode = ql)

If correct, then type **:q!** and combine bias images to form **zero.fits**
To save changes write **:wq**

apphot > zerocombine
apphot > ls z*.fits
Should show: **zero.fits**

Before running a combine for the **dark*.fits**. Check the long header for one images of the different **dark*.fits** to see the exposure time.

apphot > imhead dark1.0001.fits lo+

The following should show:
dark1.0001.fits[512,512][ushort]: dark

No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [512,512], length of user area 932 s.u.
Created Fri 10:12:17 15-Jun-2012, Last modified Fri 00:39:12 18-May-2012
Pixel file "dark1.0001.fits" [ok]

BZERO	= 32768	/ offset data range to that of unsigned short
BSCALE	= 1	/ default scaling factor
RUN	= 'dark1'	/ name of this run
OBJECT	= 'dark'	/ Object name
EXPTIME	= '5'	/ Actual integration time (sec)
FILTER	= 'BVR'	/Filter type

etc...

EXPTIME is the exposure, please remember the exposure time for that dark1*.fits. The other dark2*.fits, dark3*fits,...etc will have different exposure times.

Make sure that the following settings are in place. (Will do this for the other **dark*.fits**)

apphot > epar darkcombine

I R A F
Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = darkcombine

input	=	@dark1list	List of dark images to combine
(output	=	dark5)	Output dark image root name // dark(exptime)
(combine	=	median)	Type of combine operation
(reject	=	minmax)	Type of rejection
(ccdtype	=	)	CCD image type to combine
(process	=	no)	Process images before combining?
(delete	=	no)	Delete input images after combining?
(clobber	=	no)	Clobber existing output image?
(scale	=	exposure)	Image scaling
(statsec	=	)	Image section for computing statistics
(nlow	=	0)	minmax: Number of low pixels to reject
(nhigh	=	1)	minmax: Number of high pixels to reject
(nkeep	=	1)	Minimum to keep (pos) or maximum to reject (neg)
(mclip	=	yes)	Use median in sigma clipping algorithms?
(lsigma	=	3.)	Lower sigma clipping factor
(hsigma	=	3.)	Upper sigma clipping factor
(rdnoise	=	0.)	ccdclip: CCD readout noise (electrons)
(gain	=	1.)	ccdclip: CCD gain (electrons/DN)
(snoise	=	0.)	ccdclip: Sensitivity noise (fraction)
(pclip	=	-0.5)	pclip: Percentile clipping parameter
(blank	=	0.)	Value if there are no pixels
(mode	=	ql)	

If correct, then type **:q!** and combine dark images to form **dark5.fits**
To save changes write **:wq**

apphot > darkcombine

Make sure the following settings are in place.

apphot > epar flatcombine

I R A F
Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = flatcombine

input = @flatlist List of flat field images to combine
(output = flat) Output flat field root name
(combine = average) Type of combine operation
(reject = avsigclip) Type of rejection
(ccdtype =) CCD image type to combine
(process = no) Process images before combining?
(subsets = no) Combine images by subset parameter?
(delete = no) Delete input images after combining?
(clobber = no) Clobber existing output images?
(scale = mode) Image scaling
(statsec =) Image section for computing statistics
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise = 0.) ccdclip: CCD readout noise (electrons)
(gain = 1.) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 1.) Value if there are no pixels
(mode = ql)

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

apphot > flatcombine

Check the time exposure for the star you are working on using **imhead lo+** on one of the **<stars>.fits**.
Make sure the following settings are in place. Assume the exposure time is 10 second integrations.

apphot > epar ccdproc

I R A F
Image Reduction and Analysis Facility

PACKAGE = ccdred
TASK = ccdproc

images =	**@J1118list**	**List of CCD images to correct**
(output =	**)**	**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?**
(zerocor =	**yes)**	**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**
(zero =	**zero.fits)**	**Zero level calibration image**
(dark =	**dark10.fits)**	**Dark count calibration image**
(flat =	**flat.fits) 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?**
(functio =	**legendre)**	**Fitting function**
(order =	**1)**	**Number of polynomial terms of spline pieces**
(sample =	***)**	**Sample points to fit**
(naverage =	**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)**	

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

apphot> ccdproc

This step might take some time to complete. When it finishes you will have a set of calibrated frames.

While you are waiting, in the other terminal, you can prepare for the next step, namely aligning the images.

cp J1118list J1118alignlist
vi J1118alignlist

J1118-1.0001.fits
J1118-1.0002.fits
etc... //click on esc

Then enter:
:**%s/J1118-1/J1118-1.align/g** //Changes the name of list by using the global find and replace command in vi (editor)

The list will show:
J1118-1.align.0001.fits
J1118-1.align.0002.fits
etc...

Write and quit **vi** by clicking esc and entering **:wq**

Similarly, you can create a list for the magnitude files that you will need later.

cp J1118alignlist J1118alignmaglist
vi J1118alignmaglist

Then click esc and enter:

:%s/.fits/.fits.mag.1/g

List will show:
J1118-1.align.0001.fits.mag.1
J1118-1.aling.0002.fits.mag.1
etc...

Write and quit **vi** by clicking esc and entering **:wq**

Back to the iraf terminal, do a combine of all your raw data to see the target movement of the star.

apphot> epar imcombine

Make sure the following settings are in place.

I R A F
Image Reduction and Analysis Facility

PACKAGE = immatch
TASK = imcombine

input =	**@J1118list**	**List of images to combine**
output =	**J1118raw.fits**	**List of output images**
(header =	**)**	**List of header files (optional)**

(bpmasks =	**)**	**List of bad pixel masks (optional)**
(rejmask =	**)**	**List of rejection masks (optional)**
(nrejmas =	**)**	**List of number rejected masks (optional)**
(expmask =	**)**	**List of exposure masks (optional)**

etc...

Open image but opening display software first:

apphot> imcombine
apphot> !ds9 &
apphot> disp J1118raw.fits
frame to written into (1:16) (1): //just click enter when this line pops up

Look at the image and guess what the number of shift might be.

Now display a typical image of your star that has the best lighting and image look. Usually start with the image in the middle. Example, you have 1200 images, look at image 600. Once image is found, create a coordinate file to align all images. Pick the brightest stars or a unique pattern. Need to have a minimum of three points.

Example of a typical image: **J1118-1.0250.fits**

apphot> disp J1118-1.0250.fits
//enter the frame number you want the image in

In ds9, click on the stars you want to align with. Green circles will go on top of the star. Move your arrow to the center of the star and get the x- and y- coordinates. They will show at the top of the header.

In the other terminal, is where you create the .coo files. Enter the coordinates in a **vi** .coo file.

vi J1118align.coo

In editor enter **i**.

Example for insert:
157 296
249 392
286 372

First column is x- and second column y-. Just put a space between the numbers and click enter to go to next line.

To finish and save, click esc and enter: **:wq**

To align images, go back to iraf terminal.

apphot> epar imalign

Make sure the following settings are in place.

I R A F
Image Reduction and Analysis Facility

PACKAGE = immatch
TASK = imalign

input = @J1118list Input Images //List of images you start with
referenc = J1118-1.0250.fits Reference Image //Image used to get coord. files
coords = J1118align.coo Reference coordinates file //Coordinate file
output = @J1118alignlist Output images //Will name output images from list
(shifts =) Initial shifts file
(boxsize = 15) Size of small centering box //will change numbers if doesn't align correctly. Each star aligning is different
(bigbox = 45) Size of the big centering box //will need to change also in case images do not align
(negativ = no) Are the feathers negative?

(backgro = INDEF) Reference background level
(lower = INDEF) Lower threshold for data
(upper = INDEF) Upper threshold for data
(niterat = 12) Maximum number of iterations
(toleran = 4) Tolerance for convergence
(maxshif = INDEF) Maximum acceptable pixel shift
(shiftim = yes) Shift the images?
(interp_ = linear) Interpolant
(boundar = nearest) Boundary type
(constan = 0.) Constant for constant boundary extension
(trimima = no) Trim the shifted images?
(verbosa = yes) Print the centers, shifts,and trim section?
(list =)
(mode = ql)

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

apphot> imalign

To check combination do an **epar imcombine.**

If combine is done, but images are not aligned then enter: **del J1118*.align.*.fits**. Change settings at **boxsize** and **bigbox** at **epar imalign**. Re-run the alignment and combine until images combine.

Now you have a set of aligned calibrated images. You will use the aligned combine image from imcombine. On the imcombine image, click on your target star and three or two stars to compare the target star to. Get the coordinates of those stars and enter them into the following:

vi J1118.targ.coo //Enter the targets' coordinate and do same to compare stars.
vi J1118.com1.coo
vi J1118.com2.coo
vi J1118.com3.coo

Make sure the following settings in place (usually you only have to check this once, so if you are sure the settings have not been changed you can skip photpars and datapars).

apphot> epar photpars //enter

I R A F
Image Reduction and Analysis Facility

PACKAGE = apphot
TASK = photpars

(weighti = constant) Photometric weighting scheme for wphot
(apertur = 1,2,3,4,5,6,7,8,9,...,16) List of aperture radii in scale

(zmag = 25.) Zero point of magnitude scale
(mkapert = no) Draw apertures on the display

(mode = ql)

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

apphot> epar datapars //enter

I R A F
Image Reduction and Analysis Facility

PACKAGE = apphot
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 =	**INDEF)**	**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 =	**)**	**CCD gain image header keyword**
(readnoi =	**0.)**	**CCD readout noise in electrons**
(epadu =	**1.)**	**Gain in electrons per count**
(exposur =	**)**	**Exposure time image header keyword**
(airmass =	**)**	**Airmass image header keyword**
(filter =	**)**	**Filter image header keyword**
(obstime =	**UTC)**	**Time of observation image header keyword**
(itime =	**1.)**	**Exposure time**
(xairmas =	**INDEF)**	**Airmass**
(ifilter =	**INDEF)**	**Filter**
(otime =	**INDEF)**	**Time of observation**
(mode =	**ql)**	

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

To get the data numbers, you run the program phot for all coordinate files (J1118.targ.coo, J1118.com1.coo, …, etc). This next section deals with extracting magnitudes. Make sure the following settings are in place.

apphot> epar phot //enter

I R A F
Image Reduction and Analysis Facility

PACKAGE = apphot
TASK = phot

image =	**@J1118alignlist**	**The input image(s)**
skyfile =		**The input sky file(s)**

(coords = J1118.targ.coo) The input coordinates file(s) (default: image.coo.?)
(output = default) The output photometry file(s) (default: image.mag.?)
(plotfil =) The output plots metacode file
(datapar =) Data dependent parameters
(centerp =) Centering parameters
(fitskyp =) Sky fitting parameters
(photpar =) Photometry parameters
(interac = no) Interactive mode?
(radplot = no) Plot the radial profiles in interactive mode?

(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, world)
(wcsout =)_.wcsout) The ouput coordinate system (logical, tv, physical)
(cache =)_.cache Cache the input image pixels in memory?
(verify =)_.verify) Verify critical parameters in non-interactive mode?
(update =)_.update) Update critical parameters in non-interactive mode?
(verbose =)_.verbose) Print messages in non-interactive mode?
(graphic =)_.graphics) Graphics device
(display =)_.display Display device
(mode = ql)

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

apphot> phot

Will click enter to move to next line except in **Sky fitting algorithm**.

The input image(s) (@J1118alignlist):
Centering algorithm (centroid) (CR or value):
New centering algorithm: centroid
Centering box width in scale units (5.) (CR or value):
New centering box width: 5. scale units 5. pixels
Sky fitting algorithm (centroid) (CR or value): mode //type in **mode** and enter
Sky fitting algorithm: mode
Inner radius of sky annulus in scale units (10.) (CR or value):
New inner radius of sky annulus: 10. scale units 10. pixels
Width of the sky annulus in scale units (10.) (CR or value):
New width of the sky annulus: 10. scale units 10. pixels
File/list of aperture radii in scale units (1,2,3,...,16) (CR or value):
Aperture radius 1: 1. scale units 1. pixels
Aperture radius 2: 2. scale units 2. pixels
.
.
.
Aperture radius 16: 16. scale units 16. pixels
Standard deviation of background in counts (INDEF) (CR or value):

New standard deviation of background: INDEF counts
Minimum good data value (INDEF) (CR or value):
New minimum good data value: INDEF counts
Maximum good data value (INDEF) (CR or value):
New maximum good data value: INDEF counts

To place the extracted data you have to make sure the following settings are in place (once this is checked, settings should be the same)

apphot> epar txdump

textfile = @J1118alignmaglist Input apphot/daophot text database(s)
fields =otime, mag, merr Fields to be extracted
expr = yes Boolean expression for record selection
(headers = no) Print the field headers?
(paramet = yes) Print the parameters if headers is yes?
(mode = ql) Mode of task

If correct, then type **:q!** and combine flat images to form **dark5.fits**
To save changes write **:wq**

apphot> txdump > J1118_<date of observation>_targ.dat

//Enter to pass all lines and fully run program, should take a few seconds
Input apphot/daophot text database(s) (@J1118alignmaglist):
Fields to be extracted (otime, mag, merr):
Boolean expression for record selection (yes):

Then delete all the magnitude files:

ls *.mag.*
del *.mag.*

Then repeat section again but change:

In **epar phot:**

(coords = J1118.com1.coo

In **txdump:**

txdump > J1118_<date>_com1.dat

Again delete magnitude images. Do this for the rest of the coord. compare stars.

List all .dat files to make sure they are all done.

apphot> ls *.dat

J1118_<date>_targ.dat J1118_<date>_com1.dat
J1118_<date>_com2.dat J1118_<date>_com3.dat

You now have data files for the target and 3 or 2 compare stars.

To logout of IRAF:

apphot> logout

This will bring you to your normal terminal view once again.