How to make Color-Magnitude Diagrams of Star Clusters

Laura Chomiuk for COSMOS 2006

In order to make a color-magnitude diagram and find all the interesting information it gives, we need to first measure the brightnesses of stars in two different filters. To do this well, we will use two data reduction programs: daophot (the version for Macs) and IRAF. This manual is written for use on a Mac.

1. Photometry

Photometry is the science of measuring the brightnesses of stars. We will use a method called point-spread function (PSF) fitting. The point-spread function is the ”shape” of stars in the image. If we can model this shape well, then we should be able to fit every star with this model, subtract it out, and be left with only sky background and noise. The stuff we subtracted out is then the light from the stars– the information we're looking for!

1) Start up IRAF:

a) Start XDarwin by clicking on the Red ”X” on the dock.
b) You need to open a special kind of window called an ”xgterm” in order to run IRAF. Type ”xg
c) Go to the IRAF directory: Type ”cd iraf”. Press Enter.
d) Start IRAF: Type ”cl”. Press Enter.
e) Start up your image displayer, called DS9: Type ”! ds9 &”. Press Enter.
f) Move to the directory that contains your data by typing ”cd ../AstroClusters/”. Press Enter.

2) Display your image. This image should be biased and flat-fielded. Let's say it's called funscience.fits. Type ”display funscience”. Your star cluster should appear in the ds9 window! Play around with ds9 for a while. Particularly fun things to do are:

a) Click on ”Edit” in the menu bar, and then select ”Pan”. This will let you move around the image.

b) Click on ”Edit” in the menu bar, and then select ”Colorbar”. If you move your mouse around on the image while keeping the mouse button pressed down, you can change the way your image is displayed.

c) Click on ”Zoom” in the menu bar. This should give you the option for ”zoom in” or ” zoom out”. Push these and practice zooming in and out.

d) Click on ”Color” in the menu bar, and try displaying your image in a pretty color! Play around with this option plus that described in (b) to make your image look just the way you like it!

3) Now it's time to pick your PSF stars. These are basically ”model” stars, which should be representative of all the other stars in your image. We will use a program called *imexamine* in IRAF to record some basic information about the PSF stars.

a) Imexamine has lots of user-defined options. We want to reset these, in case any past user has been doing funky things: Type ”unlearn imexamine”.

b) Now, we want to set up imexamine for our own purposes. We'll use the *epar* command– that stands for *edit parameters*: Type ”epar imexamine”. In the first line, where it says ”input =”, type the name of your image– here, ”funscience”. Then scroll down to where it says ”logfile =”. Type ”funscience.log”, or your image name's equivalent (i.e., if your image is m11.fits, type m11.log). Next, scroll down to ”keeplog =” and type ”yes”. Finally, exit out of this screen by typing ”:q”.

c) Now, start imexamine. Type ”imexam” in the IRAF window. If you move your cursor to the DS9 window, it should now look like a donut. Your job now is very important. You need to find 25-35 rather isolated, well-exposed, but un-saturated stars. That means that a good PSF star doesn't have close neighbors. It it bright– but not too bright!

Center your donut cursor on a nice-looking star (no spikes or ”fuzz” coming from it, this would mean it was saturated!). Type ”r”. A window should come up with a radial profile of the star. Information about this star will be recorded in your log file. Keep typing ”r” on nice PSF stars until you think you have about 30. Make sure you've found some from every edge/corner of the image!

d) Type ”q” to exit imexamine.

4) Double-check your PSF stars. This part's a bit subjective. Open up your funscience.log file in a text-editor (try ”TextEdit”, which is a program that should be in the dock) Any star with ”Peak” larger than 50,000 or so should probably be deleted. Additionally, if there are any lines that don't have nice columns of numbers like the other lines, but instead have crazy words in them, delete them too!

The last two columns (”moffat” and ”direct”) let your know the width of the hill of the star. All these number should be similar, somewhere in the range of 5-7. If any stars have really large values in this column, you should probably delete them, too. I mean, delete the whole row, not just the column! Now, hopefully you'll have a nice, robust sample of PSF stars left in your .log file. You should still have 15 or so, minimum.

5) Open another xgterm window. In your IRAF window, type ”! xgterm -sb &”.

6) Prepare to run daophot on your data. Your daophot.opt file is very important to get

right. Let's edit it in TextEdit. This file determines how sensitive daophot is to faint stars. It also determines what shape the PSF is fit with. The parameters as listed below are good first guesses. However, the readout noise and gain below are not for the Nickel Telescope. You will need to look these up at:

$$http : //mthamilton.ucolick.org/techdocs/detectors/dewar2/dewar2_frame.html$$

The full width at half-maximum (FWHM) of the average star is also important to get right. You should look at the last two columns of your .log file, and use a typical number from there. Set FW, FI, and PS accordingly.

In daophot.opt:
HI=60000 (Saturation level in counts)
LO=5 (How many times brighter than sky noise a star must be for detection)
GA=2.04 (Gain in e-/DN; look up on webpage)
RE=3.1 (Readout noise in DN; look up on webpage)
PS=22 (PSF radius, set this equal to 4 x FWHM)
TH=5 (Threshold for detection. this should be the same as LO)
FI=5.5 (Fitting radius, set this equal to 1 FWHM)
WA=-1 (Watch level– controls how interactive daophot will be)
AN=1 (Geometric function for psf. 1 is a Gaussian function)
VA=1 (How much PSF is allowed to change across image.)
FW=5.5 (FWHM of point source)

The next parameter file to think about is allstar.opt. Let's edit this one to look like:

In allstar.opt:
is=30 (Inner sky radius. Should equal (4xFWHM)+(5 or 10))
os=40 (Outer sky radius, should equal IS+10, or so.)
FI=5.5 (Fitting radius, should be the same as FI in daophot.opt)
CE=0 (Some fancy clipping parameter, best ignored and set to zero)
CR=0 (Ditto)

Next, edit photo.opt. This one is the least important. It contains the options for the program phot, which basically just generates a dummy file. Type ”pico photo.opt”, and check that the file meshes with the below:

In photo.opt:
A1 = 3. (Perform aperture photometry with a 3 pixel radius)
A2 = 0 (Ignore all the other apertures, because this one is zero)

A3 = 4.6 (These other apertures don’t even matter!)
A4 = 5.6
A5 = 6.9
A6 = 8.5
A7 = 10.5
A8 = 13
A9 = 19
AA = 12
AB = 25
AC = 29
IS = 30 (inner sky radius, same as in allstar.opt)
OS = 40 (outer sky radius, same as in allstar.opt)

6) Finally, it’s time to run daophot. My friend Jenny wrote a script that does all the complicated stuff for you. The command is:

```
./doall_phot your_jmage_name number_of_images_averaged FWHM
```

So, in our example, our image is funscience.fits. We didn’t add or combine any images to make it (and neither will you), so only 1 image has been averaged. The FWHM in our example is 5.5. So the command would be: ”./doall_phot funscience 1 5.5”

Push enter, and daophot should start chugging along. It might crash if it dislikes one of your PSF stars, then you should call for Laura or Kirsten. Otherwise, it’s going to run, for a while. It will probably take 15 minutes for an open cluster, and maybe a half-hour for a globular cluster.

7) When it’s done, it will have created lots of files–including a file with the brightnesses (magnitudes) of your stars, called funscience.als. You will also have a PSF-subtracted image. This will be your original image name with a ”s” added to it– the PSF-subtracted version of *funscience.fits* is *funsciences.fits.* Check it out by going back to your IRAF window and typing: ”display funsciences”. Hopefully most of your stars have disappeared!

2. Averaged Photometry

You only need to read this section if you are reducing multiple images in a given filter! In an ideal world, you would be, because combining images will help eliminate false detections, and increase the accuracy of your photometry. The nice people at daophot have some tools that can average the information in multiple 'allstar' files (i.e. funscience.als), like 'daomatch' and 'daomaster'.

So, after running doall_phot on all the relevant images, feed the .als files for all your multiple images into 'daomatch', which will derive the shifts between the frames. These offsets are recorded in a funscience.mch file, and aren't bad, but they can definitely be improved upon, which is exactly what 'daomaster' will do. So, in the window where you were running doall_phot, now type "daomatch". It will ask your for a "Master Input File"– type in the name of one of your .als files. Next, it will ask you for a "Output File Name". You can use the default it suggests, or make one up, but the name should end with *.mch*! This file will contain the offsets between the images.Then, it will ask you for the "Next Input File". Type in your next .als file. When you push enter, lots of numbers should scroll by. Then, it will ask you for your next input file, again. If you have a third (or fourth, or fifth) .als file you wish to match up, enter it now. Otherwise, quit out of the program by pushing enter.

You can check on the offsets *daomatch* has found by reading your *.mch* file. Type "less funscience.mch" to read it. The file should list all the .als files. The two numbers directly following a file name are the pixel shifts in x and y, relative to the first image. These numbers should both be zero for the first file, as this image doesn't need to be shifted to match up with itself! If, in the other rows, the number are large (¿ 20 or so), there might be something wrong. You might want to display the images and come up with your own offsets. Ask Laura how to do this, if necessary.

Next, you need to use *daomaster* to hone in on very accurate offsets between your multiple frames, and to create a combined star catalog. Unfortunately, my *daomaster* program no longer works on Macs, so we're going to have to use a different computer for a second. Ask Laura how to do this.

When you're all set up with your *.mch* and *.als* files on the new computer, type "daomaster". *Daomaster* will ask for a "File with list of input files"; give it your funscience.mch filename, and it will then know to read in the .als files for all your multiple exposures. It will then ask you how picky you want to be about what counts as a star and what doesn't. *Daomaster* will ask for "Minimum number, minimum fraction, enough frames". Let's say you have two images you're combining. You want your stars to show up in both images, so type "2,1,2". Ask Laura about this if you want to know more. It will next ask you for 'maximum sigma'. This is another parameter that can help reject sketchy stars. Look in your .als files at the , and try to get a good understanding of what the largest error in magnitude for a reasonable looking star could be. (Magnitude errors are the fifth column; magnitudes are the fourth column). Something around this value should be your maximum sigma, usually about 0.15 or so. ’Daomaster’ will reject stars with photometry errors larger than this, i.e. saturated stars, or galaxies.

Next, ’daomaster’ asks you for the kind of offset you’d like to apply. Let’s be simple, and type ”2”. Lastly, ’daomaster’ asks your for a ’Critical match-up radius’. Type ”10”

Then starts the offset iterations! For a given match-up radius, ’daomaster’ calculates the offsets for stars, and gives you errors in these offsets. Keep typing that same match-up radius until the number of stars in the master list repeats itself several times in a row.

So, for example, if *daomaster* prints some lines that look like this:

```
1.26 1.29 0 0 0.00 0.015 2612 2 cr164_b.als
2,508 stars within radius 10.00
New match-up radius (0 to exit):
```

2508 is the number of stars in the master list. If you keep typing ”10” and pushing enter, this number will dance around a bit, and then should steady. This means that the computer has matched the images up well within an accuracy of 10 pixels and excluded false detections, and now you can ask it to get even more accurate. When the number of stars starts repeating itself, shrink the match-up radius by one (start typing ”9”). Keep doing this until your match-up radius is ”1”. When you think your offsets are as accurate as they are going to get, type ’0’ for your match-up radius. *Daomaster* will ask if you’d like assign new star IDs. Say ”yes”. Next, *daomaster* now offers to spit out all sorts of fancy stuff. Of course, you’d like to see the new and improved offsets, so say yes to ’a file with the new transformations’. ’Daomaster’ also knows how to average the star brightnesses in the various .als files correctly, and these magnitudes will represent the nicest photometry you’re going to get for your data, so definitely say yes to ’A file with mean magnitudes and scatter?’. You’ll get a funscience.mag file, which will represent your final photometry for that field in that filter!

3. Colors

Ultimately, you should have a final .mag file for both B and V filters. So, now we can generate colors, and eventually, color-magnitude diagrams! In order to get colors, run all the above steps twice, one for each filter. So, let’s say we observed funscience in B and V; we now have our averaged B-band photometry in funscience_b.mag, and our averaged V-band photometry in funscience_v.mag. Now run ’daomatch’ and ’daomaster’ on these two fields to match them up well. However, when it comes time to choose your output from 'daomaster', you of course don't want the averaged photometry, as that would be a garbled mess of the two filters. Instead, say yes to 'A file with raw magnitudes and errors?'. Call it something like 'color_funscience.mag', and it will contain your B and V magnitudes, all nice and matched up.