

Image reduction and analysis at the Peter van de Kamp Observatory

D. Cohen and E. Jensen

(v.1.2 November 30, 2014)*

Software

The software we use to reduce (i.e., apply calibrations to) data is *AstroImageJ*. We also use it to combine images and to analyze photometric variability (measure a star's relative brightness over the course of many observations) of stars that may host transiting exoplanets.

The software is freely available:

www.astro.louisville.edu/software/astroimagej/

It runs on Macs, PCs, and under Linux.

These notes are based on v.2.1.5 (but with the daily build update applied, so formally it's v.2.1.6db). It's generally a good idea to check the version you're using and update it if an update is available. Choosing the "daily build" option (from the version pull-down menu) will get you the most up-to-date version, sometimes beyond the most recent numbered version.

Data Reduction

The basic unit of observing is a single image. Color images of extended objects (like planets, nebulae, and galaxies) are made by combining separate images each taken with a different color filter. And exoplanet transit light curves are made from a sequence of images that contains the host star (not usually taken in more than one filter).

Images are produced by our CCD detector: A Charge-Coupled Device (CCD) is the common type of electronic light sensor in nearly all cameras these days, including camera phones. Astronomical CCDs are bigger, have smaller pixels, and have better (lower) noise properties than standard commercial CCDs. When light hits a CCD it induces a charge in the pixel in which it lands. The raw data of an image is a map of the charge in each pixel. You can forget the physics and think of the image as an array of integers with values proportional to the amount of light that landed on that pixel during the time the CCD was exposed to light. But the raw data also contain noise from the detector itself. Before we analyze our data or make our images, we need to remove as much of this noise as possible.

* Added plate solution instructions to add WCS information into the header, enabling alignment to be done automatically.

The reduction of most observations is similar. Three types of corrections are made to each image: a bias correction which removes the more or less constant signal caused by the voltage applied to the CCD – a sort of “zero point” – that you get even without exposing the CCD at all; a dark current correction which accounts for the thermal noise (charges induced in pixels by heat rather than light; we cool our CCD typically to -20 degrees C to minimize thermal noise, but you can’t eliminate it); and a flat-field correction that corrects for pixel-to-pixel sensitivity variations (so that a uniformly bright source of light would induce the same signal in each pixel). Each of these corrections is made by empirically measuring the thing you’re trying to calibrate (the bias level, the thermal noise, and an image of a uniform light source) and then subtracting or dividing out that noise estimate.

The bias is measured by taking many zero-second long exposures, and then averaging the resulting images. The dark current is measured by taking many exposures that are the same length as the data observations, and averaging them together. And the flat field is measured by taking many exposures of a patch of the dusk sky (“sky flats”), and averaging them together.

To summarize, there are three different types of calibration files that you apply to the data via *AstroImageJ*:

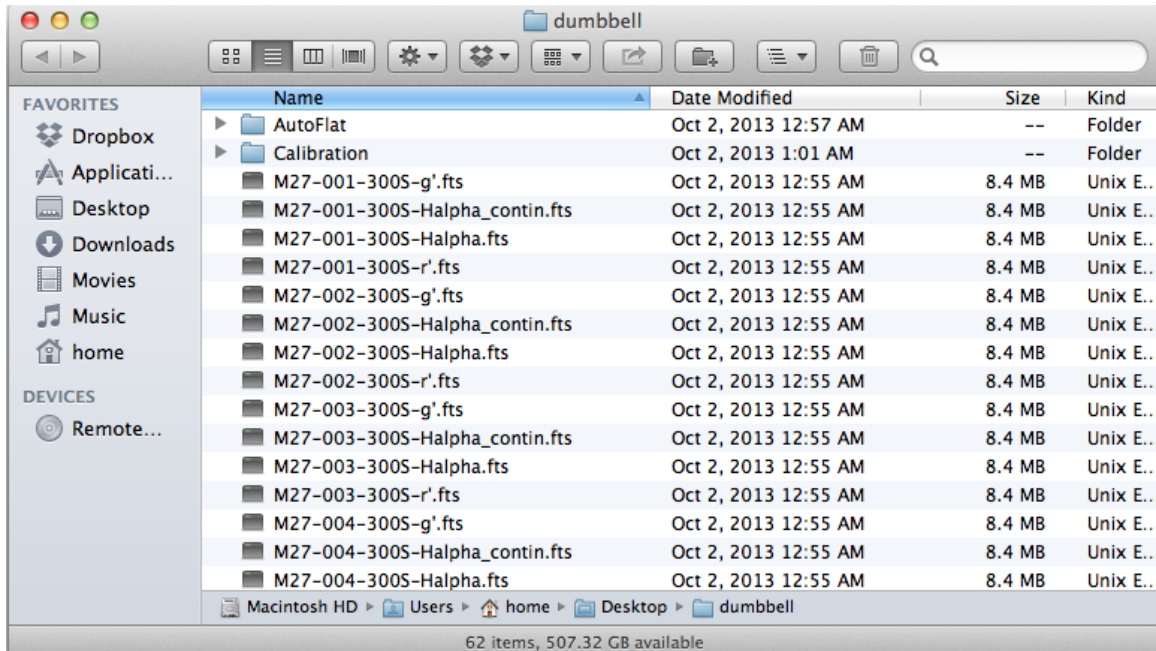
1. Bias frames (in the *Calibration* folder): these are zero-second long exposures that measure a kind of “zero-point” of voltage applied to the detector;
2. Dark frames (in the *Calibration* folder): these are exposures the same length (i.e. detector on for the same amount of time) as the data, but the camera shutter is never opened. These measure the noise induced by heat – rather than light – in the detector; and
3. Flat frames (in the *AutoFlat* folder) that are images of the dusk sky taken to measure the pixel-to-pixel variation of the detector sensitivity. There are separate sets of flats for each filter.

To get an idea of what raw images taken with the telescope and CCD camera look like, and how the data reduction and image combining works, let’s download the archive of images of Messier 27 (M27), also known as the Dumbbell Nebula.

<http://astro.swarthmore.edu/~cohen/telescope/dumbbell.zip>

When you unpack the archive, you’ll have sixty images (15 in each of four filters) and two folders (Calibration and AutoFlat). The AutoFlat folder has several images using each of the four filters (indicated in the file names, just as for the data images). The Calibration folder has 31 bias images, and darks with various exposure times (indicated in the file names).

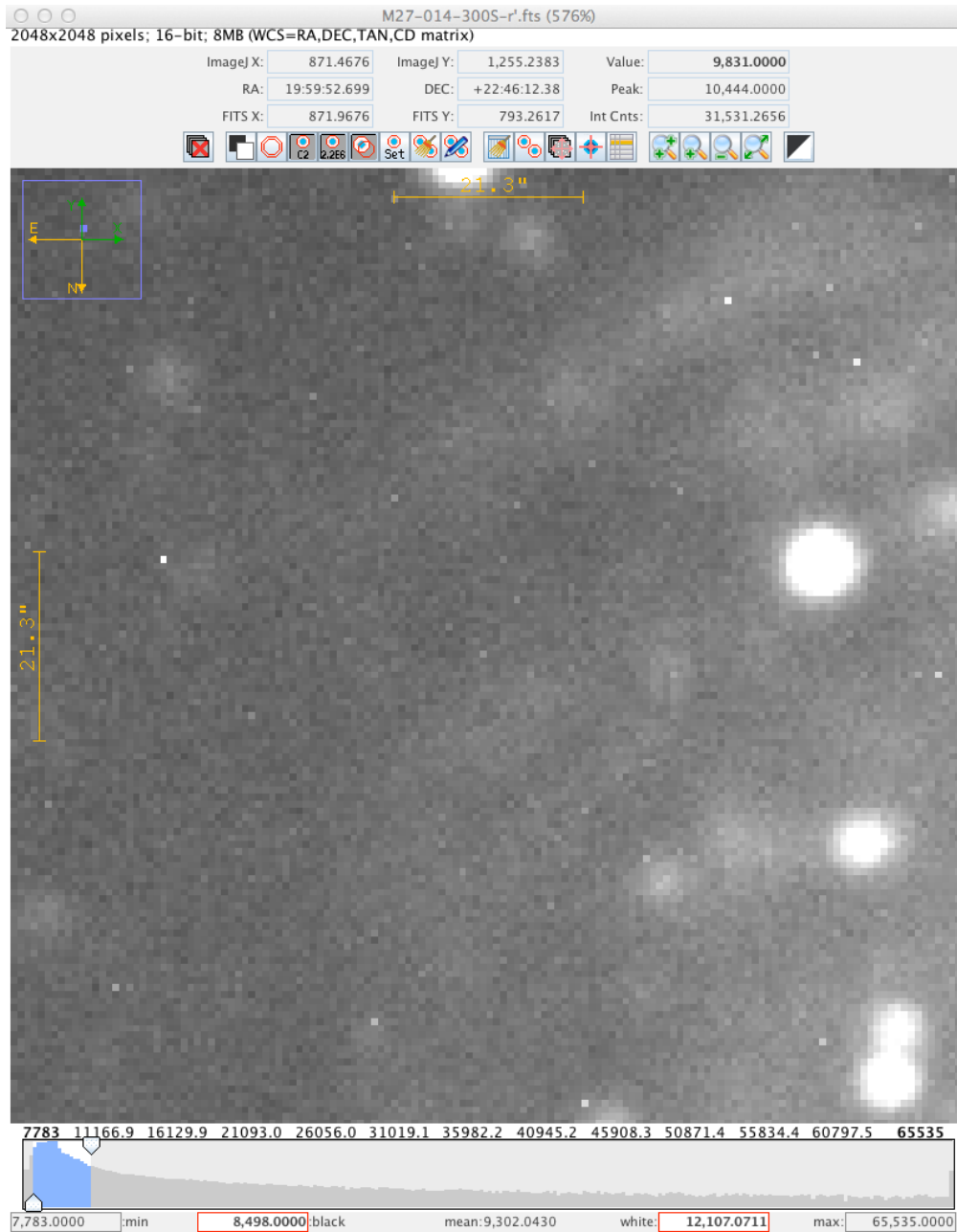
Here’s what your unpacked archive will look like:



Explore a bit before you reduce the data. Open up a representative source image in *AstroImageJ*. First, note that there are 15 300-second-long observations in each of four filters (so, sixty image files in FITS format (.fts), which can be read by *AstroImageJ*). The filters are called: g', Halpha, Halpha_continuum, and r'. Use the *File > Open* menu to open one of the g'-filter images. Then do the same for r'-filter, H-alpha, and H-alpha-continuum images (all from the same number observation, ideally). Note that as you mouse over an open image, some numbers at the top change. The "Value" is the number of counts in the pixel the cursor is over (proportional to the amount of charge in that pixel); it is what determines the shade of gray that's shown in that pixel (whiter means higher numbers).

Note also that with an image in the foreground, the *Edit > FITS header* menu shows you the so-called header (full of descriptive information - metadata! - about the image; see, e.g., the exposure time (300 seconds) on line 9).

Try changing the grayscale in one of your open images using the horizontal sliders near the bottom of the window. Zoom in on a star. See how many pixels are in any one star's image? Note that the only thing that gives the image of the star perceptible size is the blurring effect of the Earth's atmosphere (and to a lesser extent in this case, the optics of the telescope). The true size of any star is much, much smaller than a single pixel.

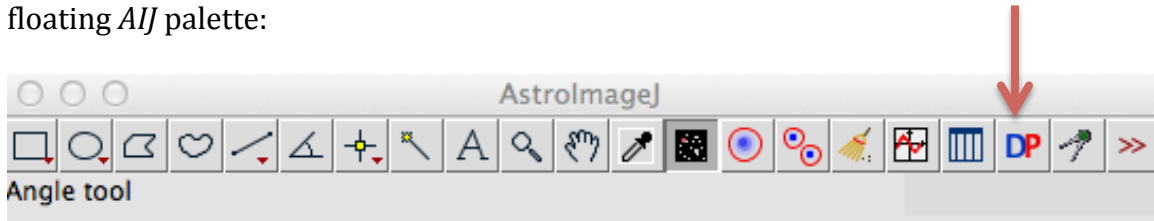


Now we can **reduce** the data. Conceptually, the workflow is: compute “master” (i.e. average or median and calibrated) bias, dark, and flat frames (one flat for each filter if you have data taken in more than one filter), and apply each of these corrections to each data frame (the word “frame” is often used to denote a single image file). However, since some of the calibration frames themselves need correcting, the specific order is (1) produce the master (average) bias frame, (2) average the dark frames and then subtract the bias from the averaged dark frame to produce a master dark frame, (3) average the flat frames for each filter, then subtract the

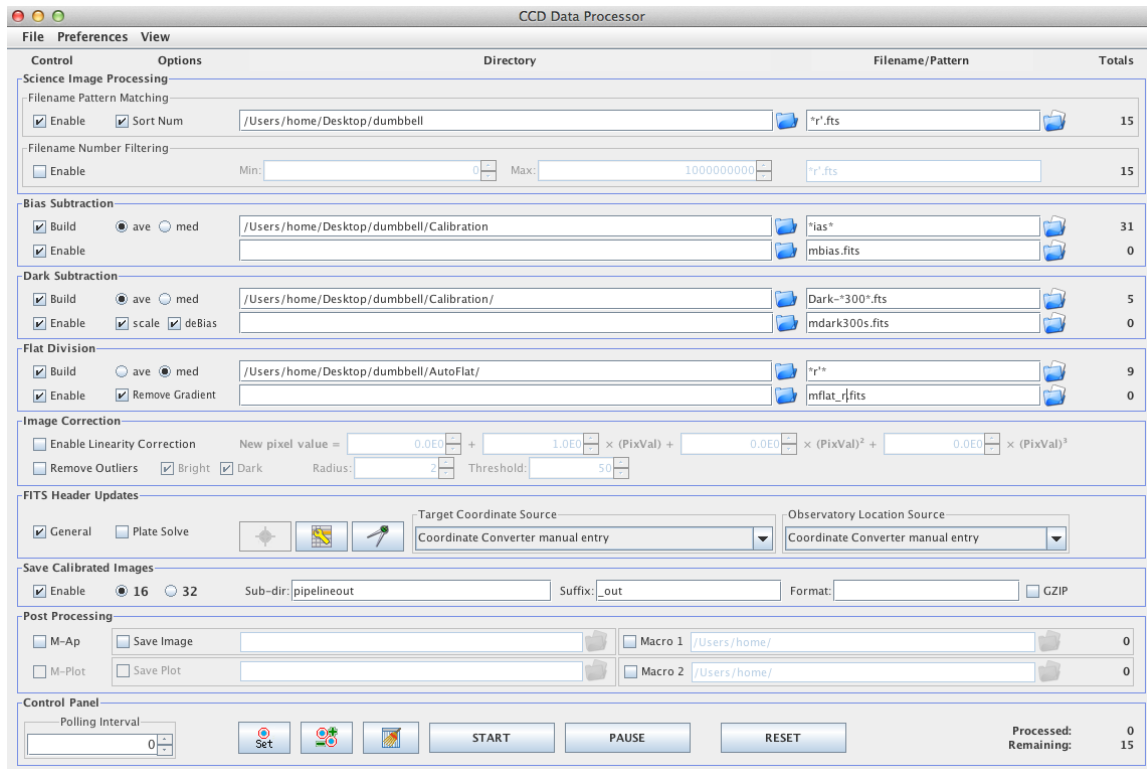
master dark and master bias to produce a master flat frame for each filter, (4) apply all three masters to each data frame.

In practice, *AstroImageJ* (AIJ) can do this in one single step (albeit with a lot of settings having to be set correctly), or at least one single step for each filter.

First, we want to open up the CCD Data Processor window. Click on the “DP” in the floating AIJ palette:



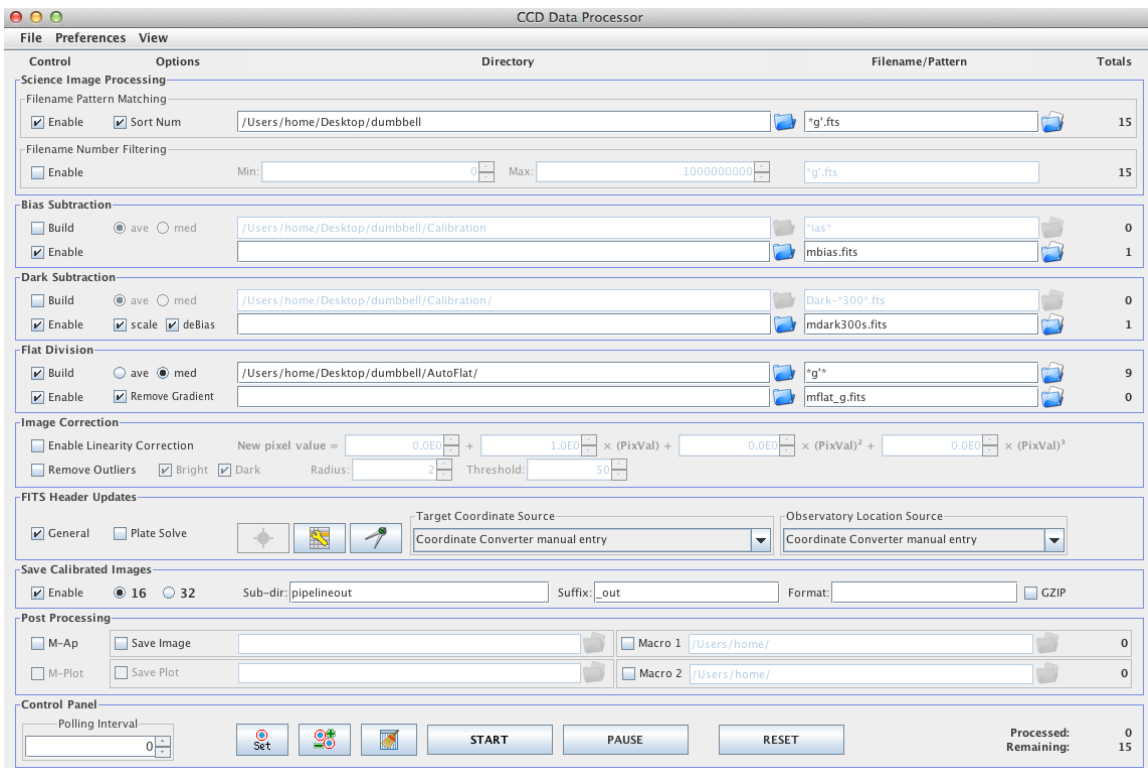
You can close the DP Coordinate Converter window. In the CCD Data Processor window, indicate the path to the folder with the data files near the top, and use the wildcard character (*) to identify all data frames taken with a given filter. Fill out the bias, dark, and flat sections, too, pointing in each case to the correct set of files. Check “build” in each case to make the master file, and click “enable” so that master gets used. Here is what the interface looked like for the r’ filter data:



A few things to note: The top part of the interface is about the input files. You have to set the path (set of nested folders) to each category of files (data frames at the top, the bias, dark, and flat below that). The “filename/pattern” windows are handy for picking out a subset of files in a given folder; I find the “Totals” column to be very useful for indicating when I’ve correctly specified the desired files (because I usually know how many such files I have in a given folder; and on the flipside, if the number of input files is “0” you know you’ve probably specified some file or folder name incorrectly. Note further that for the bias, dark, and flat sections, checking the “Build” box will make the master file in each case and checking the “Enable” box will apply that master subtraction to the data frames. (Once you’ve built, say, a master bias frame, you don’t need to rebuild it for the second set of images taken with another filter).

After clicking the “start” button you’ll see messages about the steps that are being run. Then the reduced files (just for the r’ filter) are put in a folder called “pipelineout”. Open one up and compare it to the un-reduced version (in the “dumbbell” folder). You should see much less of a diffuse glow and its spatial variation in the reduced image compared to the raw image.

Next, reduce the g’ filter images. You can reuse the master bias and master dark you’ve already made (uncheck the “build” buttons), but you do have to build a new master flat with the g’ filter:



Repeat this process for the H-alpha and H-alpha_cont filters. It's tricky in this case to get the wild-card naming of files correct, while excluding those files you don't want.

Now you should have sixty new files in the pipeline folder. These are your reduced data. Directions for loading them in and examining them are given on the next page.

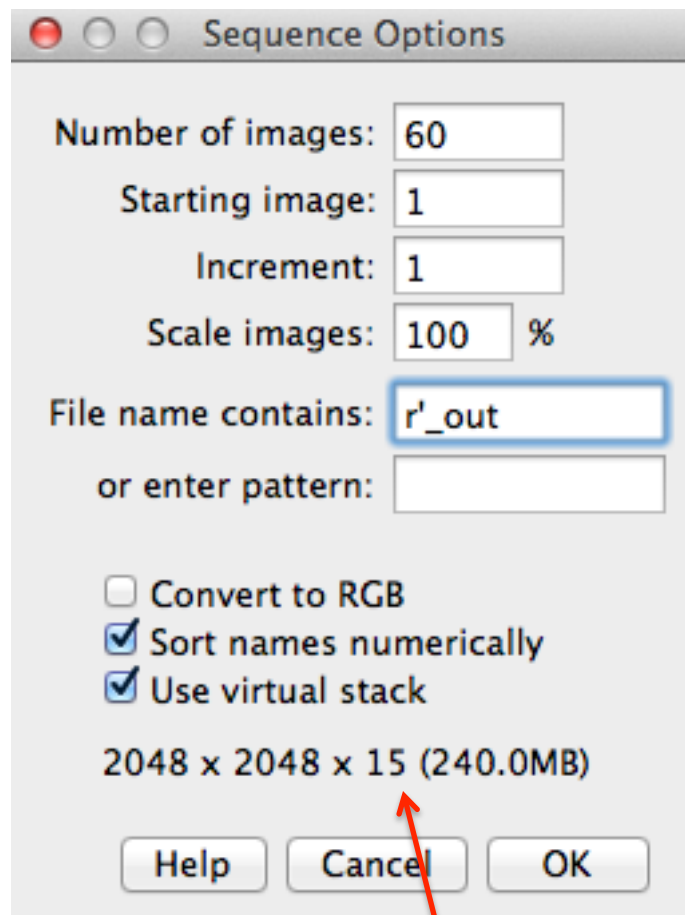
At this point, the integer values in each pixel should ideally contain information only about photons coming into the telescope from the sky. But (a) there will always be some residual noise that isn't completely calibrated out, (b) dim sources have statistical uncertainties (because you only measure a few photons, and just like an opinion poll that only polls a few people, you may not get a truly representative response), and (c) some of the photons from the sky are noise (light from very distant stars and galaxies that happen to be almost exactly behind the thing you're trying to observe – astronomical backgrounds – and also light pollution; artificial light scattered off of clouds or haze in the atmosphere – sky backgrounds). So, you should keep these additional sources of error/noise in mind, but after reducing your images, you've got a well-calibrated image in which each pixel's value is a good representation of the relative brightness, or flux, of the part of the sky imaged onto it.

Image Processing

Load in a set of images all taken with the same filter. Here we'll look at the 15 images of the Dumbbell Nebula taken with the r' filter.

Start up *AstroImageJ*, and go to File → Import → Image Sequence.

Navigate to the folder with your reduced science frames. A box will come up, with parameters like “Number of images,” “Starting image,” etc. Go to the box that says “File name contains:” and put in part of the filename that uniquely identifies the files you want (and excludes others). For instance, if you want to open up all the images taken with the r filter, all those images end in “r'_out.fts” in the folder, so you can put “r'_out” into that box. Make sure to click the “Use virtual stack” clickbox.



Note the “15” near the bottom (I’ve pointed a red arrow at it). It tells you that your “file name contains:” filter has picked out 15 files (which is correct in this case). Checking this number is a good way to ensure you’ve chosen the correct files.

Now all the images should be in a stack. You can flip through the images with the left and right arrows, and there's also a bar towards the bottom of the window that you can drag to change the picture. The name of each picture should be towards the top left of the window, so you can check that the images are really the right ones.

You can delete any images that look bad (e.g. out of focus) at this point. The icon with the red X will delete images from the stack.

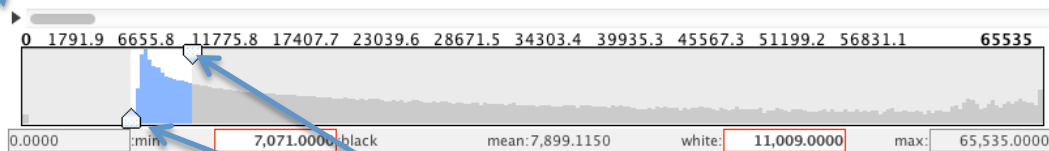
alignment icon

pipelineout (V) (35.6%)
 1/15 (M27-001-300S-r_out); 2048x2048 pixels; 16-bit; 120MB (WCS=RA,DEC,TAN,CD matrix)

Image X:	2,068.1890	Image Y:	1,944.5288	Value:	0.0000
RA:	19:58:46.393	DEC:	+22:54:57.20	Peak:	7,659.0000
FITS X:	2,068.6890	FITS Y:	103.9712	Int Cnts:	0.0000

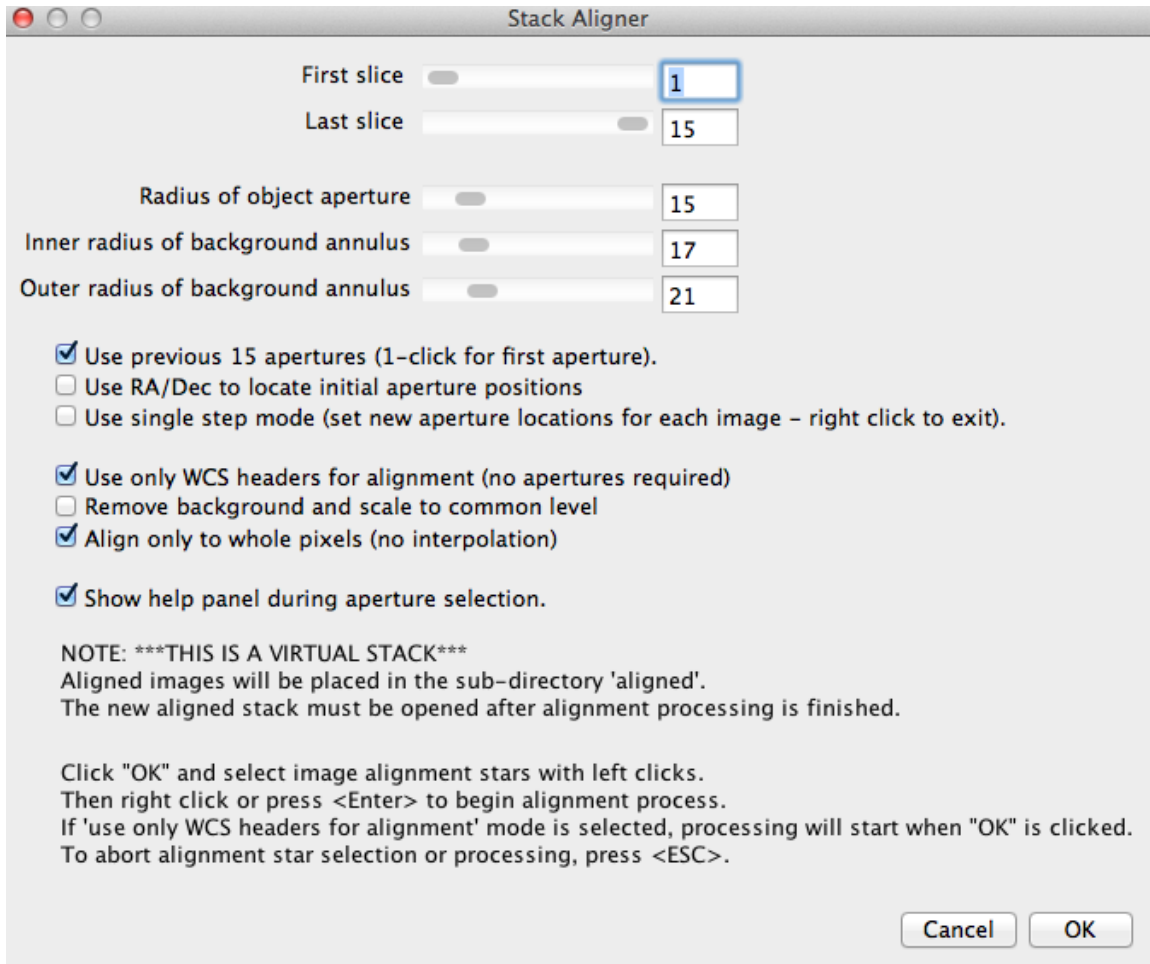


flip through images



control the grayscale

Now, we want to align the images and then add (or average) them together into one master image (in this particular filter). With the stack of images open in *AstroImageJ*, click on the alignment icon above the image (arrow at the top of the image on the previous page points at it). This dialog will open:



Note that it's easy to accidentally not set the first slice to 1. Make sure the First and Last slice options are really set to the values you want.

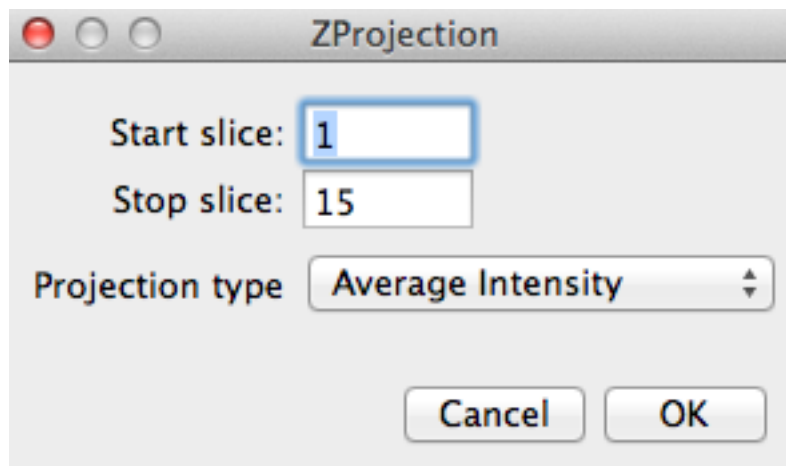
If the "Use only WCS headers for alignment (no apertures required)" box is checked, then most other options are overridden and the alignment of your opened images is done using the coordinates of the center of each image, as determined by checking the stars seen in each images against a catalog of stellar positions). This method is very reliable, and we generally should use it. But if you uncheck that box, you can align the images using a process that you have more control of. If you want to explore that option, then uncheck that box (but keep all the other settings as indicated in the screen shot above).

To do the alignment manually, you would left-click on a selection of stars (your choice) in the image. Pick about a dozen, distributed around the image. Choose stars that are neither very bright nor dim and which aren't blended with other stars. After you've chosen about a dozen, right click anywhere in the image. *AIJ* will align the images (based on the stars you've chosen) and create a new folder in which it will put the aligned images. *Again, this procedure, described in the previous paragraph, is optional. It is generally better to just use the automatic image alignment based on the so-called WCS headers (WCS = world coordinate system).*

You can close the stack and now import the newly created images in the folder called "align."

Note that the Dumbbell nebula images were generally quite well aligned to begin with, so you won't notice much difference when you flip through the aligned images, although the first g' image is offset from the others, and that clearly gets corrected in the alignment process.

Now we can average the good aligned images in the stack of all the images in a given filter (in the *align* folder). Open these aligned frames using *File > Import Image Sequence*. From the menus at the very top of your computer screen, choose *Process > Combine stack slices into single image*. The Average Intensity option is fine:



Then save this single, averaged, aligned image using *File > Save image/slice as FITS*.

Repeat the process (alignment, read in aligned images, average them, save new master image) for each filter.

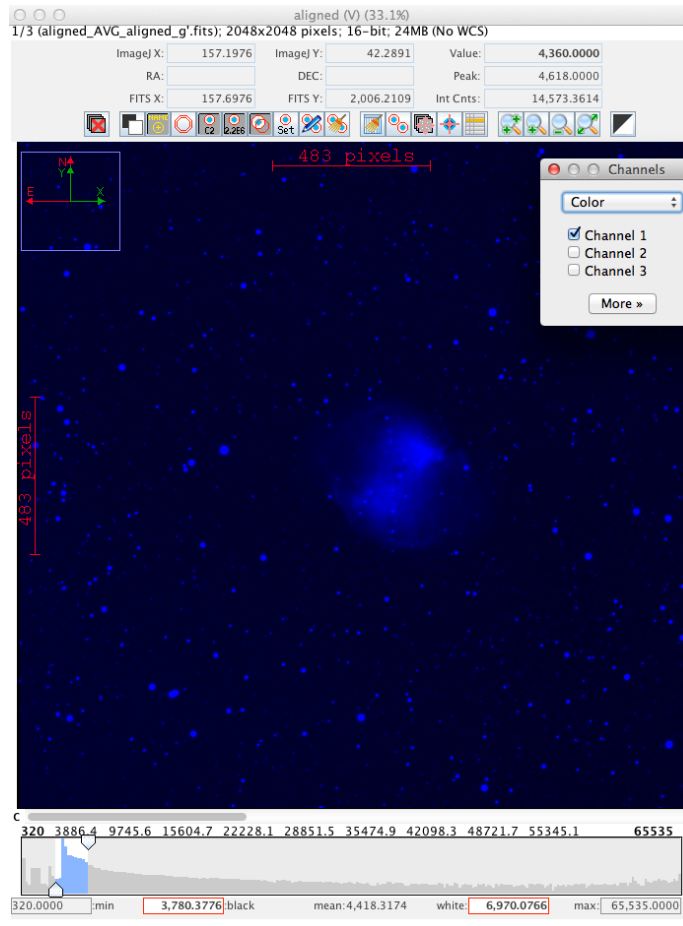
At this point, you should have four master images, one for each filter.

Open these master images in a stack (same way you've done it before: *File > Import > Image sequence*). For the Dumbbell data, this would be four images: r, g, Halpha, and Halpha_contin.

These master images may still need to be aligned with each other. To do that, you will have to use the method in which you choose the apertures by hand (described at the top of the previous page); the automatic alignment won't work on an image that's been created from a combination of other images. Alternatively, you can plate solve the images and (automatically) enter that information into the image file headers enabling you to use the same automatic WCS method for the alignment. Instructions for this are after the filter information, on p. 16. Once you've done that – by either method – the four aligned master files will be in the folder `dumbbell/pipelineout/aligned/aligned`.

Open the aligned master files in a virtual stack.

In the window containing your stack, go to the menu at the top of the screen *Color* → *Make Composite color image*. A box will come up; use the drop-down to change it so that it says “Display Mode: Color.” Then another box will come up, with the title “Channels” and the check box for Channel 1 selected. What we're after here is to assign the correct color to each channel before combining them. *AstroImageJ* doesn't know in advance which channel is which, so initially it will assign the colors red, green, and blue to channels 1, 2, and 3 respectively. If those aren't the right colors, then you can change them. When you check the box for a given channel, the image will be displayed, and you can see from filename at the top of the window which channel it is; use the button labeled “More »” to choose the appropriate color for that image. Do the same for channels 2 and 3 (and 4, if you've loaded in four frames to the image stack). Here's the g' master image in blue:



At this point you can also use the grayscale sliders to adjust the brightness of each channel. In the case of our Dumbbell Nebula dataset, the r' filter is red, and the g' filter is blue-green, so assign them to red and blue, respectively. We don't have any true green filter, but the Halpha continuum filter turns out to work pretty well for filling in that gap (at least in terms of making decent-looking images), so assign it to green. If you opened the Halpha master image at this point, go ahead and delete it from the stack.

Once you've assigned the correct color to each channel, you can see the combined results of your work by changing the display mode from "Color" to "Composite". Ideally the resulting composite image should show relatively white stars and relatively black sky. You can go back to color mode, and adjust the grayscale sliders (or even choose a different display color for a given master frame) and remake your composite image.

To save the combined channels as a single image use the "More" button and the "Convert to RGB" option. This will open a new window with a single, combined color image. Then you can save the combined image in any number of formats using the file menu. Here is my combined image in jpg form:

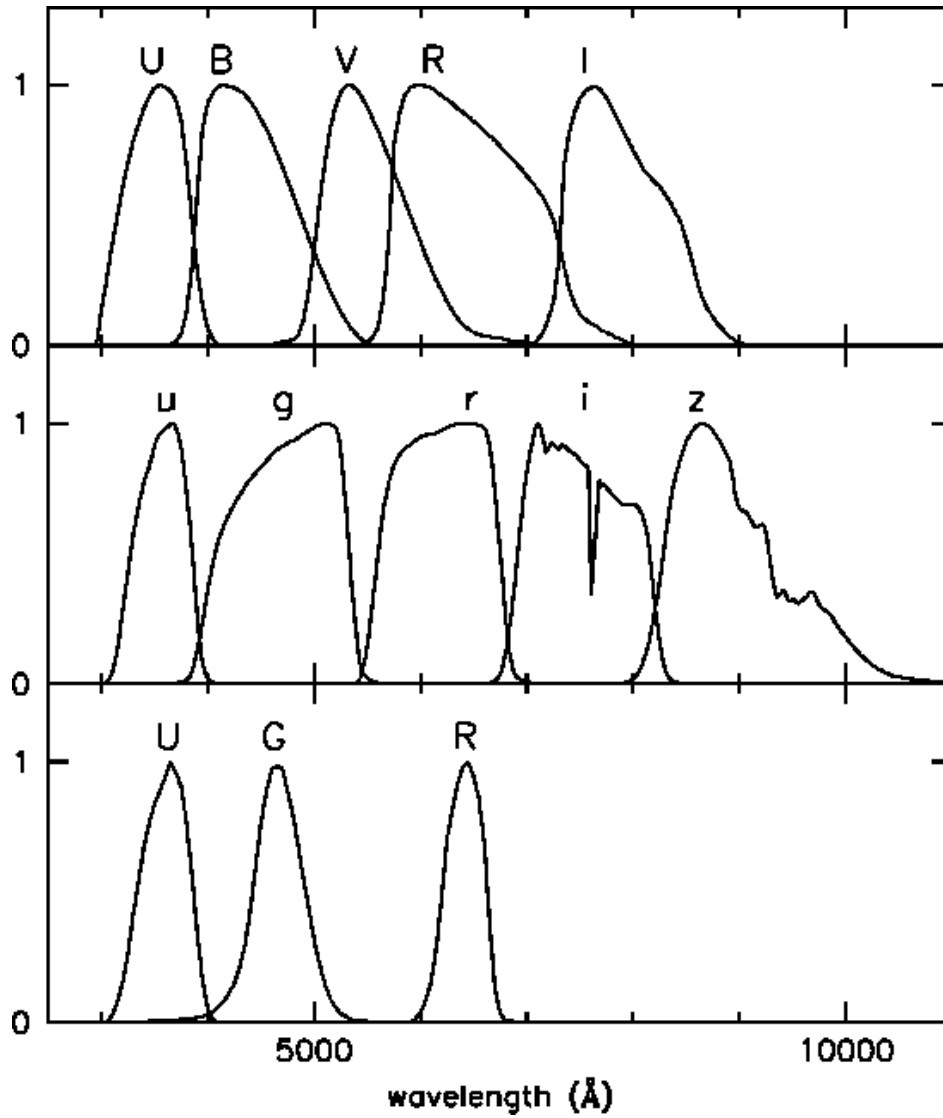


You can go through the process again with these observations of the Whirlpool Galaxy, M51:

<http://astro.swarthmore.edu/~cohen/telescope/M51.zip>

A Note About Filters

We have a set of Johnson filters (top row, in figure below) and SDSS filters (middle row), though right now (September 2014) our Johnson V filter is damaged and unusable. This makes it difficult to sample the yellow/green part of the spectrum (between 5000 and 6000 Angstroms, roughly). The B, V, R combination is the classic blue, green, red filter set. For the Dumbbell we use g' , H $_{\alpha}$ -continuum, and r' and it produces decent results, but ideally we'd use V for our green channel. The H $_{\alpha}$ filter is narrow (about 30 Angstroms wide) and centered on the strongest optical line of hydrogen, at 6563 Angstroms (quite red to the eye), while the H $_{\alpha}$ -continuum filter has two windows on either side of the H $_{\alpha}$ filter, both a little broader than 30 Angstroms.



The SDSS g filter (which we've called g' in these notes) is actually quite blue (left). While the V filter (right) is much greener.

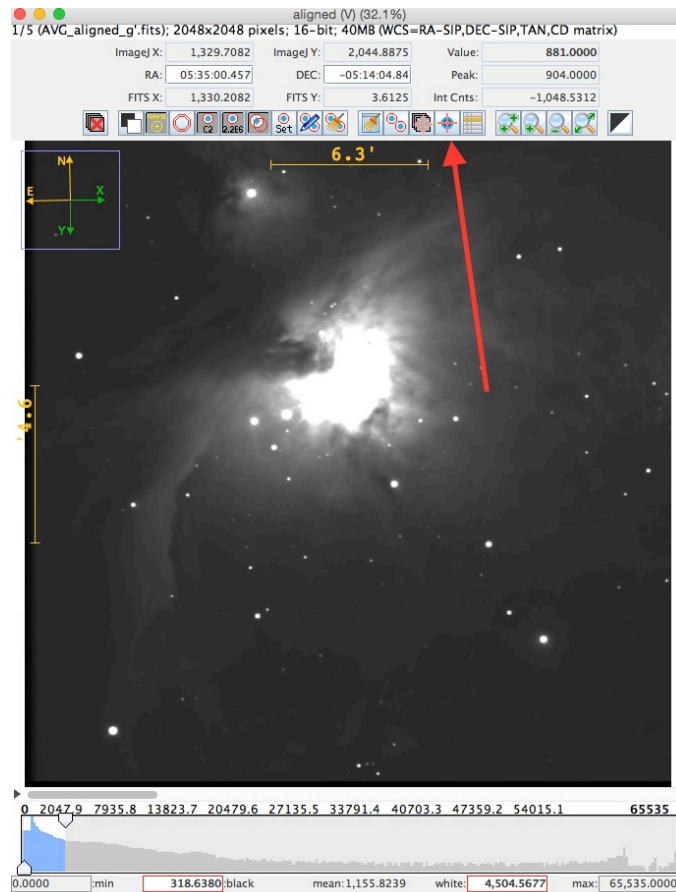


Plate solving images to put WCS information in their headers

This could be useful for aligning images when the aperture method doesn't work. For example, near the end of the process you have one master image from each filter, but since they're each made by aligning separate images, there's no longer any valid information in the image file header about the coordinates of the center of the image. That's the information that enables the alignment process to be run automatically, without having to draw apertures around numerous alignment stars. If you can put the coordinates back into the header, you can just use the same alignment procedure you use for the individual files (checking the WCS box).

To get those coordinates - you have to actually "plate solve" each image, which means: identify stars, find the pixel coordinates of the center of each star, query a database of star positions and find patterns that match what's in the image, use that to impose a new coordinate system on the image that's based on astronomical coordinates (RA (right ascension) and Dec (declination)), put the coordinate value (RA, Dec) of the center of the image into the header.

Luckily, you can do that in essentially one step. Use the plate solve button that's just to the right of the alignment button.



You can use the default settings in the menu that pops up. If you've got your virtual stack of five images loaded, then it'll do all the images at one time. Note that it takes quite a few seconds to do each image - be patient while it runs, but you can watch it find stars - drawing circles around them - and then it flashes up names/IDs and resets the scales at the top and left of the image from pixels to arcminutes. And it sets the North-East arrows correctly and colors them yellow. Once the process finishes, you should be set to go - you can align the images in the usual way. Note that the plate solving does not create new images, it just updates the images you have open.

Note: the first time you run the plate solving, you'll have to register at the astrometry.net website. Astrometry is the process of finding positions, or coordinates, for stars. AIJ uses astrometry.net behind the scenes to do the plate solving. So...to do that, you have to go to: <http://nova.astrometry.net/> and create an account (click on "sign in") and once you've done that, go do the dashboard and to "my profile" where you'll see a key printed in green text. You type that key into the AIJ interface and you can run it.