Using PyMovie / PyOTE to Process Your Asteroid Occultation Recordings – for “Team Santa Cruz”

Richard Nolthenius

Cabrillo College Astronomy, and

Earth Futures Institute, UC Santa Cruz

This resource is for Santa Cruz's **Asteroid Occultation Team**

- However, newbies everywhere will probably find something of value too.
- I keep a webpage for every asteroid occultation attempted, and it has my own observations, and that of my team members as well.
- I also have an annual summary website page which links to each page.
- My goal is to be able to instantly go back to any event and find out what I did and if there's any issues needing addressing, and also as a good record for others, like students, to see how this science is done.

Other Links of Use for New Asteroid Occultation Observers…

- https://groups.io/g/IOTAoccultations/topics this is the International Occultation Timing Association's very active message board for observers and all things asteroidal.
- I make a public webpage with light curves etc for every event our team tries. I also have a few sentences summary for each, and for 2025 that's here
- https://www.dr-ricknolthenius.com/events/past/past25.html
- My 'Events' page is my day planner, and mostly has asteroid events. At the top is links to MANY useful places
- https://www.dr-ricknolthenius.com/events/evindex.html

PyMovie / PyOTE Installation on a PC

- None of what I show here may apply for Mac's. I'm not a "mac person" so won't be able to help you on that. I assume you have a Windows 10 PC. Probably Win11 works the same.
- I assume you have installed the software above. But briefly, if not, this is a 4-step process. Super brief condensation… (see IOTA web for details)
- 1. **You Install the Python Programming Language**. This is free, but does take some steps. Instructions link at the IOTA site
- 2. **You Install PyMovie**
- 3. **You Install PyOTE**
- 4. You look for the latest update(s) and install those, using the command ">pip install==pymovie 3.9.5" (or whatever is the latest version), on a CMD line in Windows

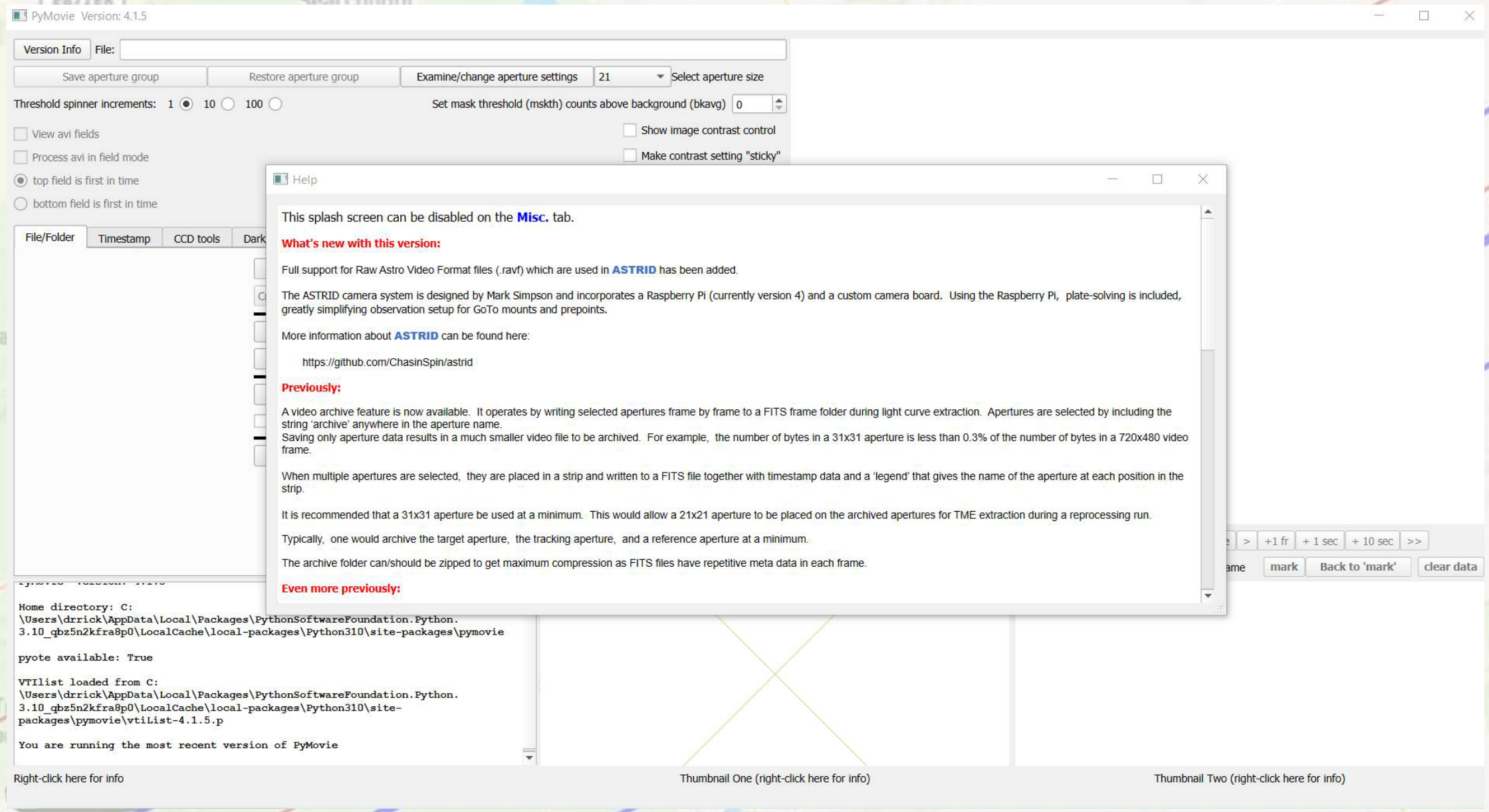
First, I click on my .*avi* video of the event, saved on my external HDD, so it'll play.

- This should open it up in Windows Media Player, or something similar.
- I play it, which has an audio channel too, and jot down on paper …
- 1. The start UT time to nearest second
- 2. The integration setting from my live comments, any other settings and observing conditions, exact description of where I was…
- 3. My location, in case I've forgotten, so later I can go to Google Earth and get my long/lat/elevation
- 4. The end time of the recording

Next, I click on my Desktop Short-Cut to Open PyMovie (image below is part of my Win10 desktop)

- It'll take a few seconds for your computer to launch Python and you'll see some command line things flow by,
- And then you'll see the opening screen for PyMovie…

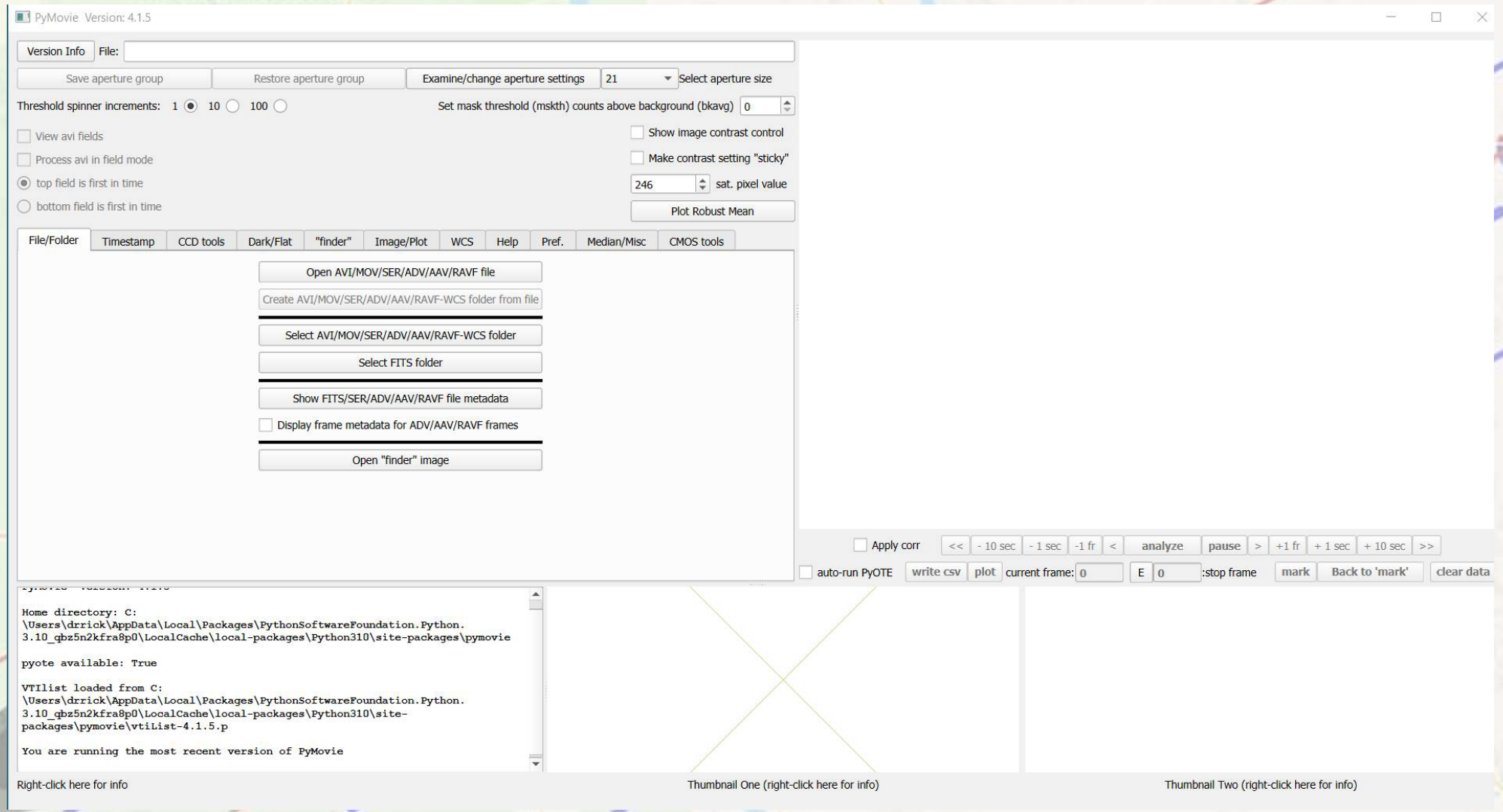
Here's the Opening Screen for PyMovie as of Jan 2025



“x-out” the “help” box, and then we get to work…

<u>Your first click</u>, is at the top of the choice boxes in the middle:

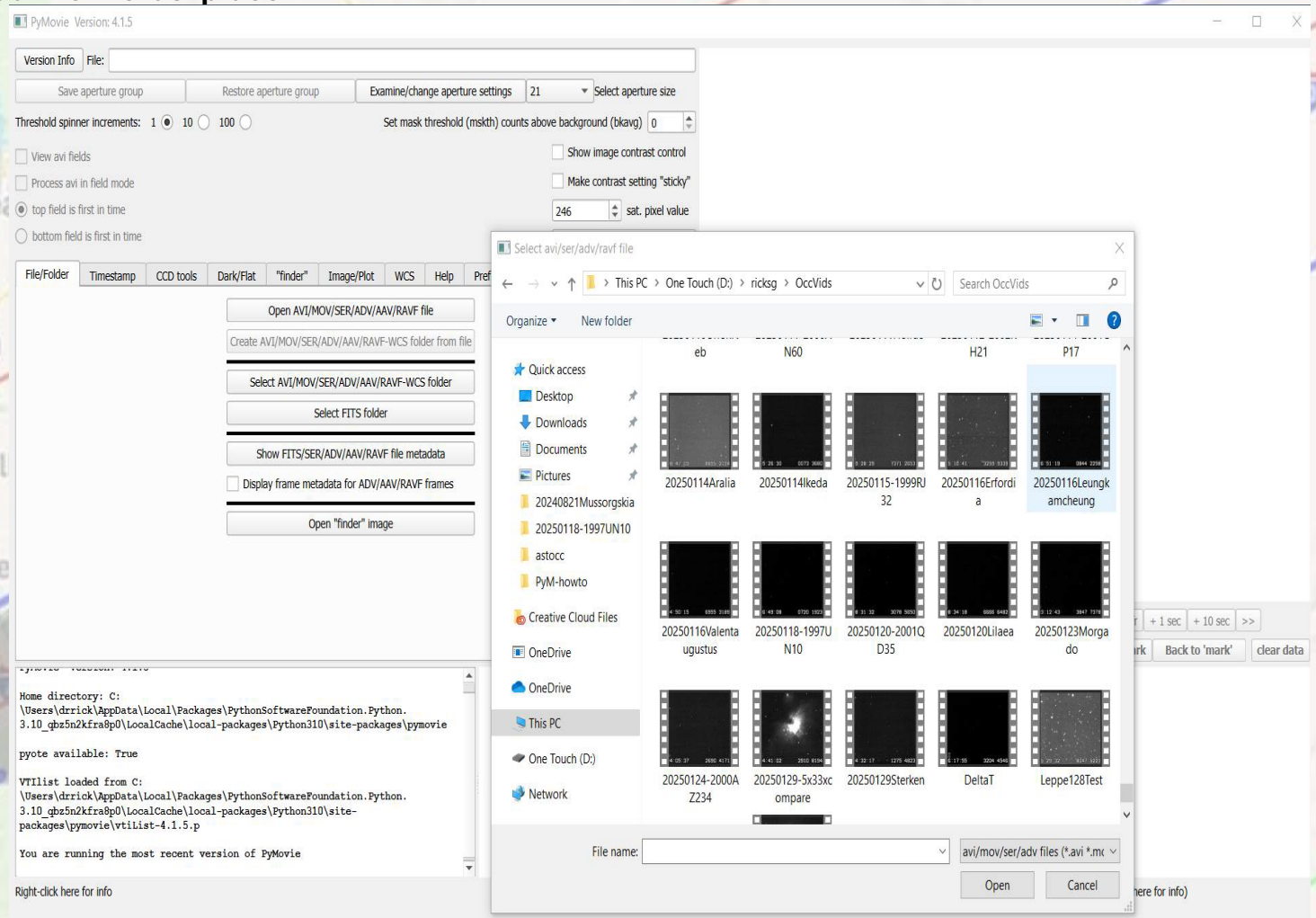
***“Open AVI/… File”*. That will bring up a box for you to navigate to where your video file is (next slide)**



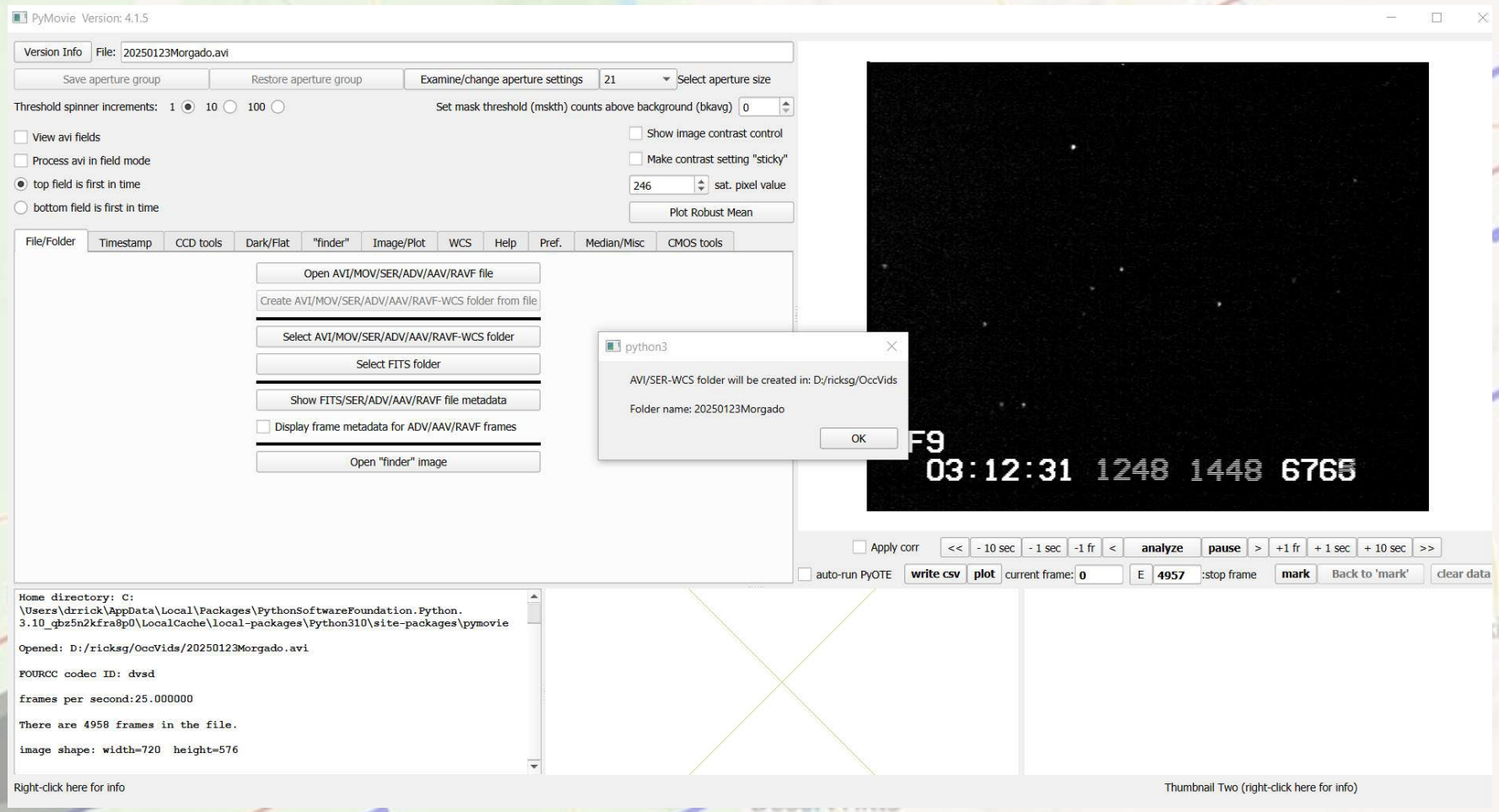
For an example, I'm going to reduce my video for the Morgado occultation of Jan 22/23, 2025

Navigate to where your raw .avi videos are. It should default to the last place

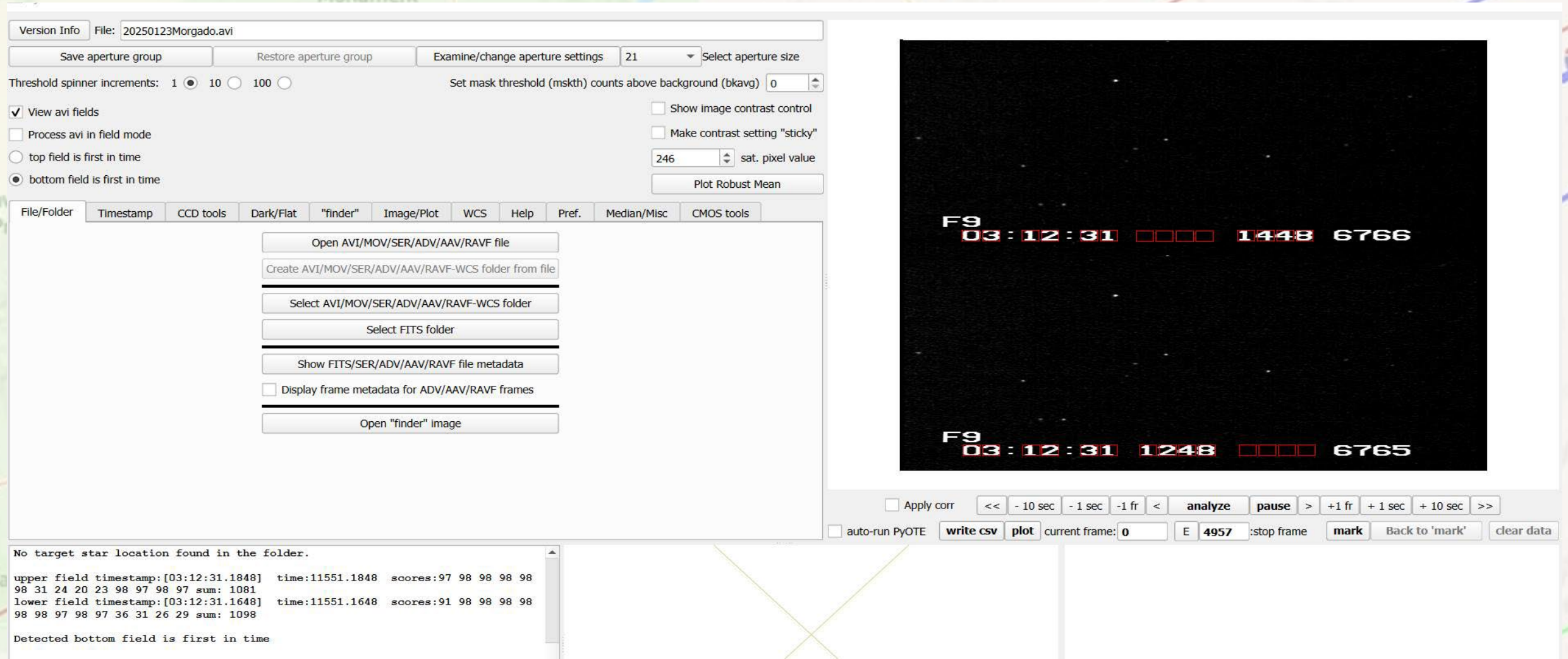
- I keep all my videos on an external HDD as they take up far more disk space than I want to use on my internal PC hard drive.
- That means that the software will also default to deposit its new files and new folders on the external HDD too, which is indeed what we want.



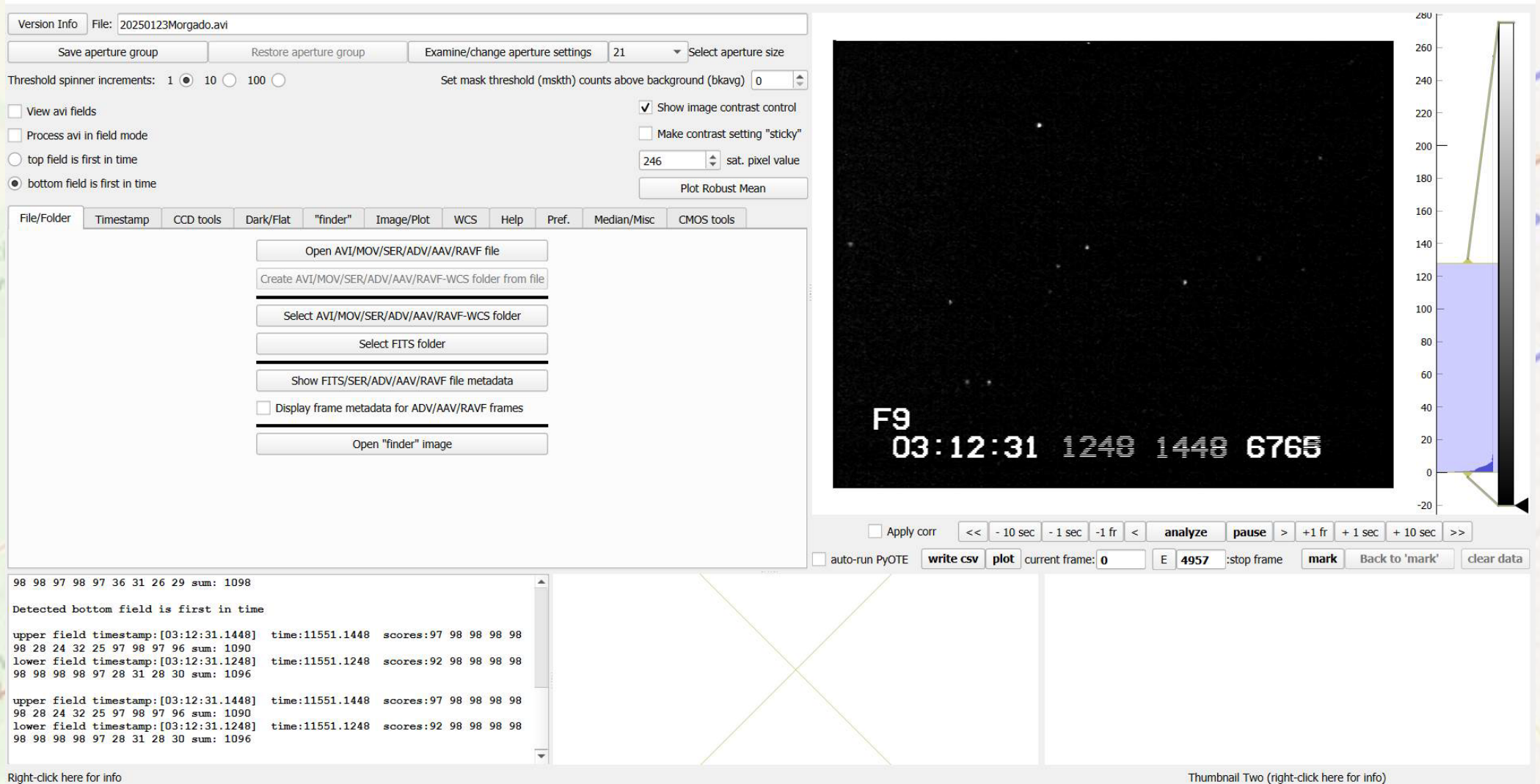
Click on your .*avi* to open it. The little box in the middle informs you it will now create a folder to hold PyMovie's and PyOTE's results; by default in the same folder that has your .*avi* file – which is what we want. Just click on "ok" and it goes away



Then it gives this screen. You'll see each of the two fields that constitute the first frame of the video: Our *.avi* frames consist of two "fields"; one field= the even rows of pixels, the other field= odd rows. It should automatically note that the bottom field has the earlier UT time (see on video the fractions of a second are earlier (.1248) than the top field's .1448) and fill the "bottom field is first in time" that you see on the left of the PyMovie screen below.



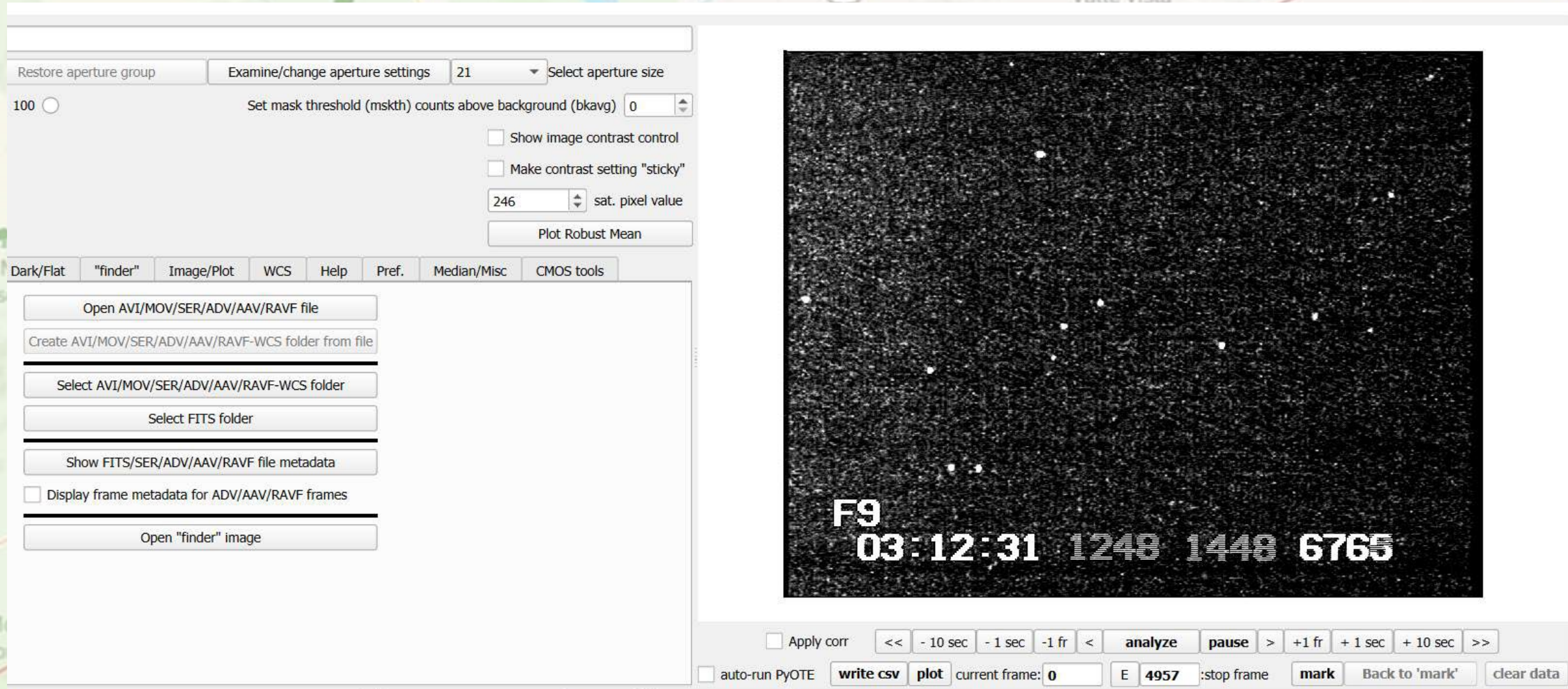
Unclick the *"view avi field"* button at far left and click on *"show image contrast control"* (middle) and you'll see view below, showing the first full frame… Note each frame has a UT time stamp (bottom left information box); it has successfully OCR'd your VTI timing numerals. (If not, you see UT=0's (zeros) and you'd need to OCR-train PyMovie)



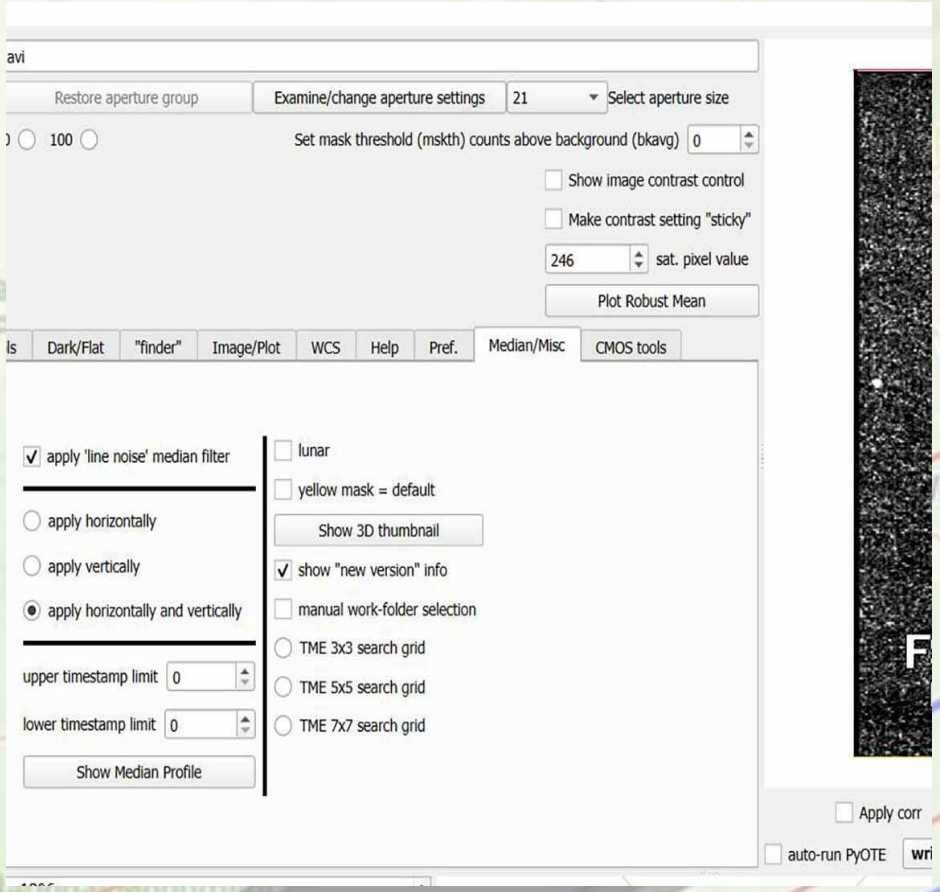
If you need to OCR-train PyMovie…

- … you need to do it just once for a given IOTA VTI, then you can save the “profile” and it’ll just automatically use it for future reductions.
- To OCR train your PyMovie, go to the IOTA website explaining and linking to videos on how to do that. Basically you move around boxes until they sit squarely on your numerals, and tell it what example number is in each box. You have to tell it what “1” is, “2” is, etc through “9” on your particular VTI. If you don’t have a VTI but have some other way to time stamp your frames, then you can do a “manual time stamp” on two of your frames later, in PyOTE. It’ll interpolate/extrapolate the other frames and put times on all of them – in PyOTE.
- If you need to do a “manual time stamp” in PyOTE, then just let the UT=0’s remain for now. So… we go Onward…

Now click on the contrast vertical graph on right side of last slide's image, and use your mouse to grab and adjust the top and bottom levels to maximize visibility of your stars. It may increase sky noise too. In order for this new contrast setting to stick, you immediately have to Unclick the *"show image contrast control"* box. When you do, the vertical contrast bar will disappear, as it has below. Don't contrast so far as to saturate needed pixels.



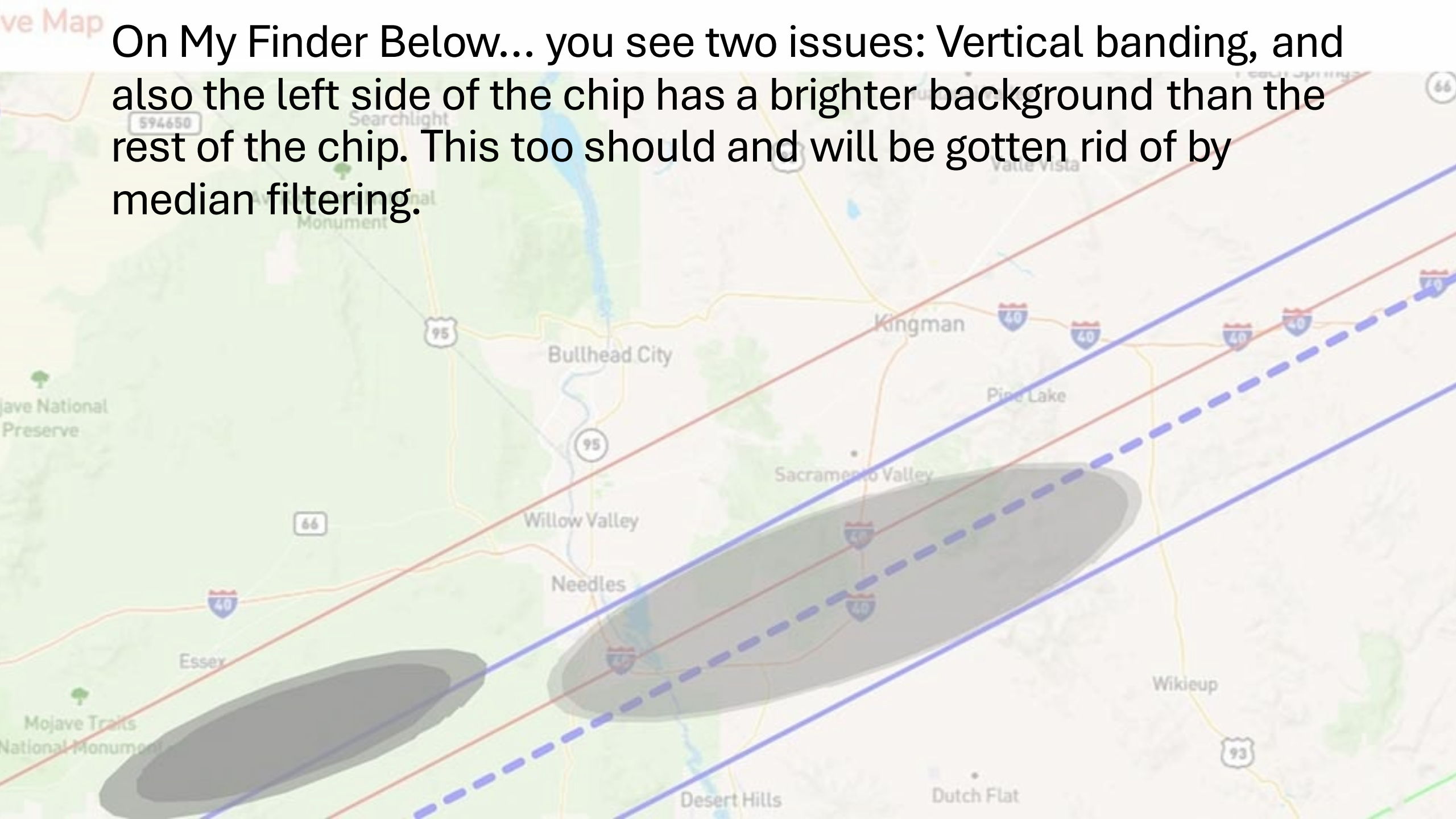
Now look at the row of tabs along the middle of the control panel, and click on the ***"median/misc"*** tab. Then click on the ***"apply line noise median filter"*** and below that, "***apply horizontally and vertically***"



- Why? Because the Watec 910hx uses separate amplifiers for each one of the hundreds of columns of pixels, and they don't all amplify identically. This means the pixel intensities even on a perfect flat field, will show vertical and/or sometimes horizontal banding due to varying sensitivity.
- The median filtering will do a flat field in a way which prevents any dead (value=0) pixel clusters from divide=ruining the photometry result.
- **I regard this step as now essential for maximizing the signal/noise for my own and others, for our Watec's**

To See the Vertical or Horizontal Banding…

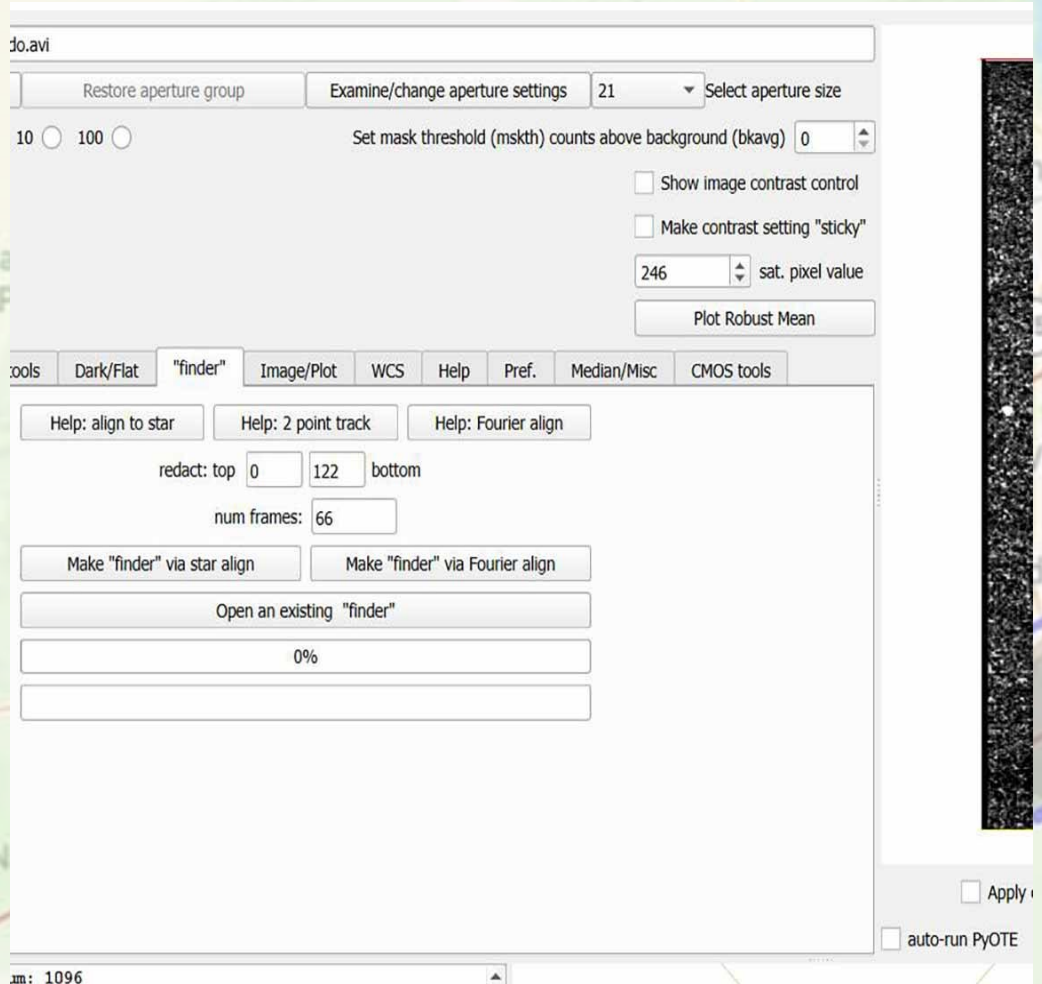
- Inside PyMovie do a “Fourier Finder” and look at the result. The displayed image will now be very smooth in the background sky, as the stacking of frames averages out the frame-to-frame random read-out and sky noise that is also an artifact and not real. That “snow” of background is noise. It is random noise.
- The “systematic errors” are what we are trying to now remove. The most obvious in my Watec is the vertical banding. It may be subtle in your finder. To show it better, adjust the contrast bar by pulling up the bottom and pulling down the top.


On My Finder Below… you see two issues: Vertical banding, and also the left side of the chip has a brighter background than the rest of the chip. This too should and will be gotten rid of by median filtering.
Searchlight
Monument
Kingman
Bullhead City
Pine Lake
Willow Valley
Needles
Essex
Mojave Trails
National Monument
Wikieup
Desert Hills
Dutch Flat

What the median vertical filtering does is this:

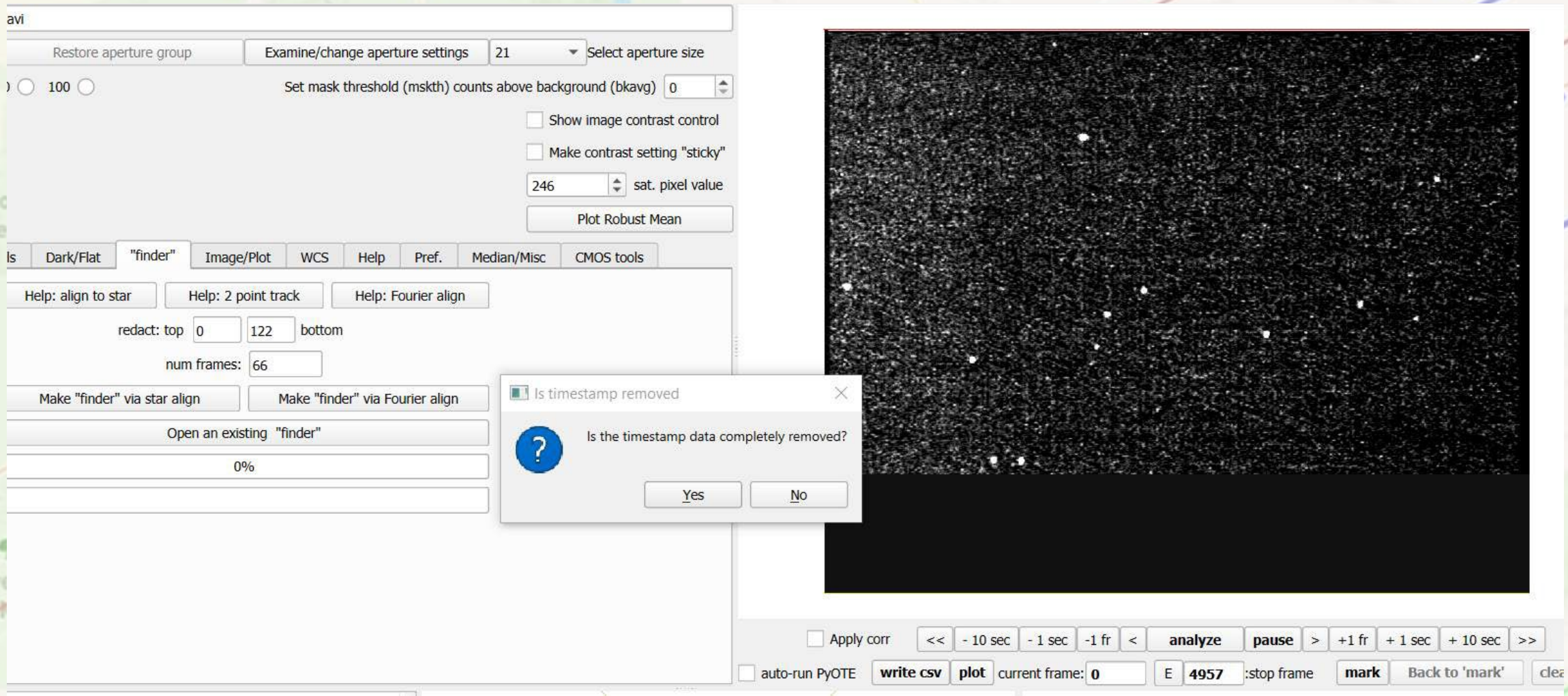
- It finds the median value of all the pixels in a column, for all columns. It then finds the median of all those median column values. It then divides each of those column medians into the median of all columns.
- It then uses the correction factors thus generated, to correct each pixel. It thus removes the vertical banding by enforcing that each column amplifier now be operating the same.
- This is the correct thing to do. There is no imaginable reason why a legitimate flat field should show vertical banding if perfectly executed with perfect pixels. The real sky does not DO vertical banding, nor horizontal either.
- This process is not blurring over “reality”, as one person felt it might be doing. It is forcing the imperfect hardware and firmware to represent a key aspect of obvious reality – that vertical or horizontal banding is a false artifact and not real, so don’t wreck your data by letting the imperfections of the hardware/firmware let such banding remain unchallenged.

If your target or reference stars are faint, you'll want to place your apertures on a "finder" field. A "finder" is basically a stacked set of frames which minimizes the sky noise seen on individual frames.



- Click on the "finder" tab.
- Click on "redact" and use 0, 111, or perhaps 0,100 as the field rows to remove from consideration. We want to exclude the numerals and only see stars.

It'll ask you if the time stamp is removed. You see in my example, the answer will be *'yes'*. Fiddle with your redact numbers until the time stamp is removed but not much above the time stamp is removed. Not critical how many frames to stack. A few dozen is fine. Then click '*Make "finder" via Fourier Align*". This will then spend a few seconds stacking frames and…



…show you the result, which should be a very smooth sky except for the banding, and perhaps the left side being brighter, as we talked about. Those will be corrected for when you do ‘analyze’ and the median filtering you specified takes effect. You may want to click on the *‘show image contrast control”* again and adjust that to maximize the visibility of your stars. Unclick *“show image contrast”* and the control at far right will disappear and the contrast view stay.

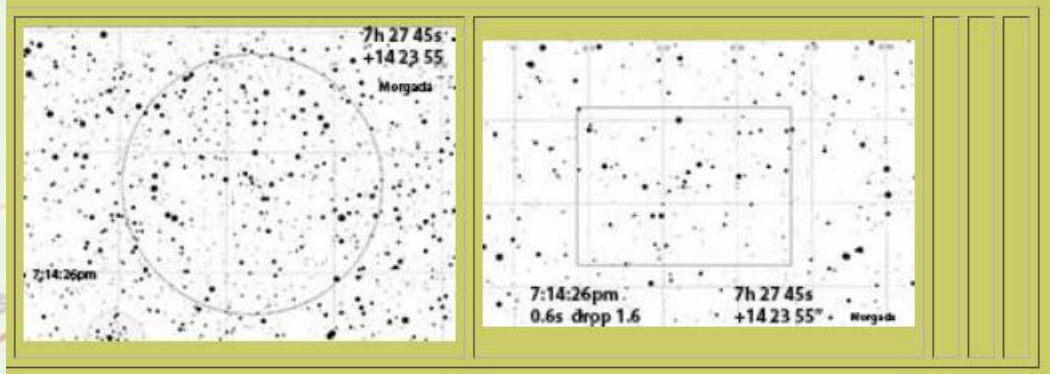
Now it's time to click on your selected stars… first, I go back to my own "planning page" which shows the Watec chip image of the field, (chart "LCD"; the square one) so I can see where my target star+asteroid is.

The Occultation of a W=11.8 Star by Asteroid Morgada

Wed eve Jan 22, 2025 at 7:14:26pm

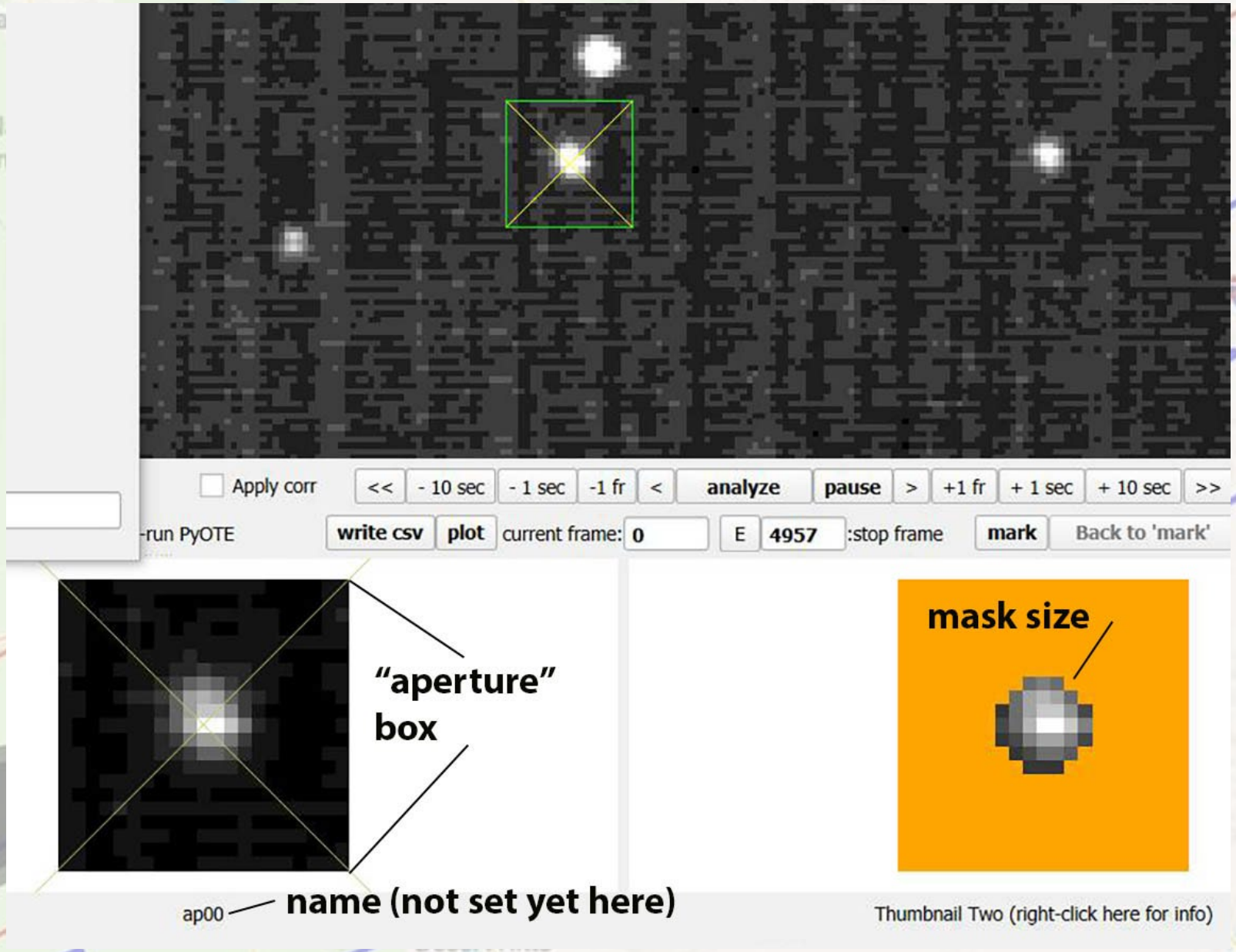
OWc page

This is a decent event. Bright enough that the 0.6s duration should be detectable without much trouble. Try for 2x if po
can get inside the path at the SC Lighthouse, and almost as good is Natural Bridges Rd. But the duration will no doubt



Results:

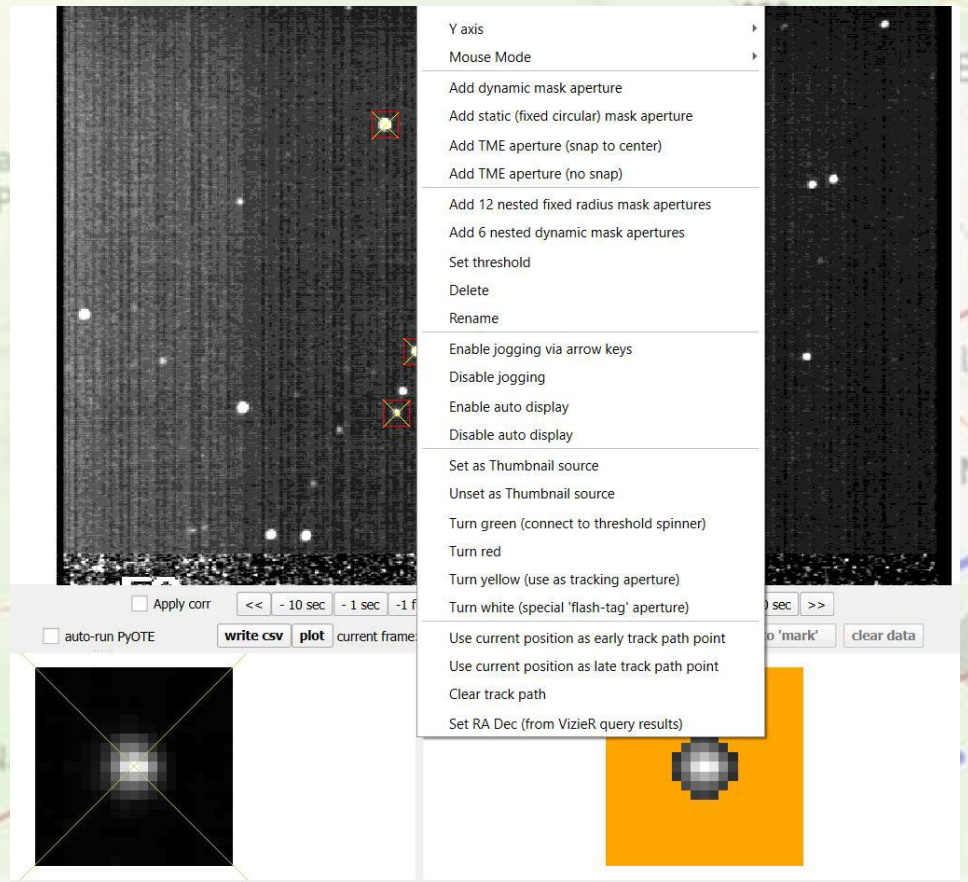
What is “Aperture” *vs.* “mask”? – Can be confusing…



Selecting aperture (box) size and mask (which pixels will measure starlight) sizes…

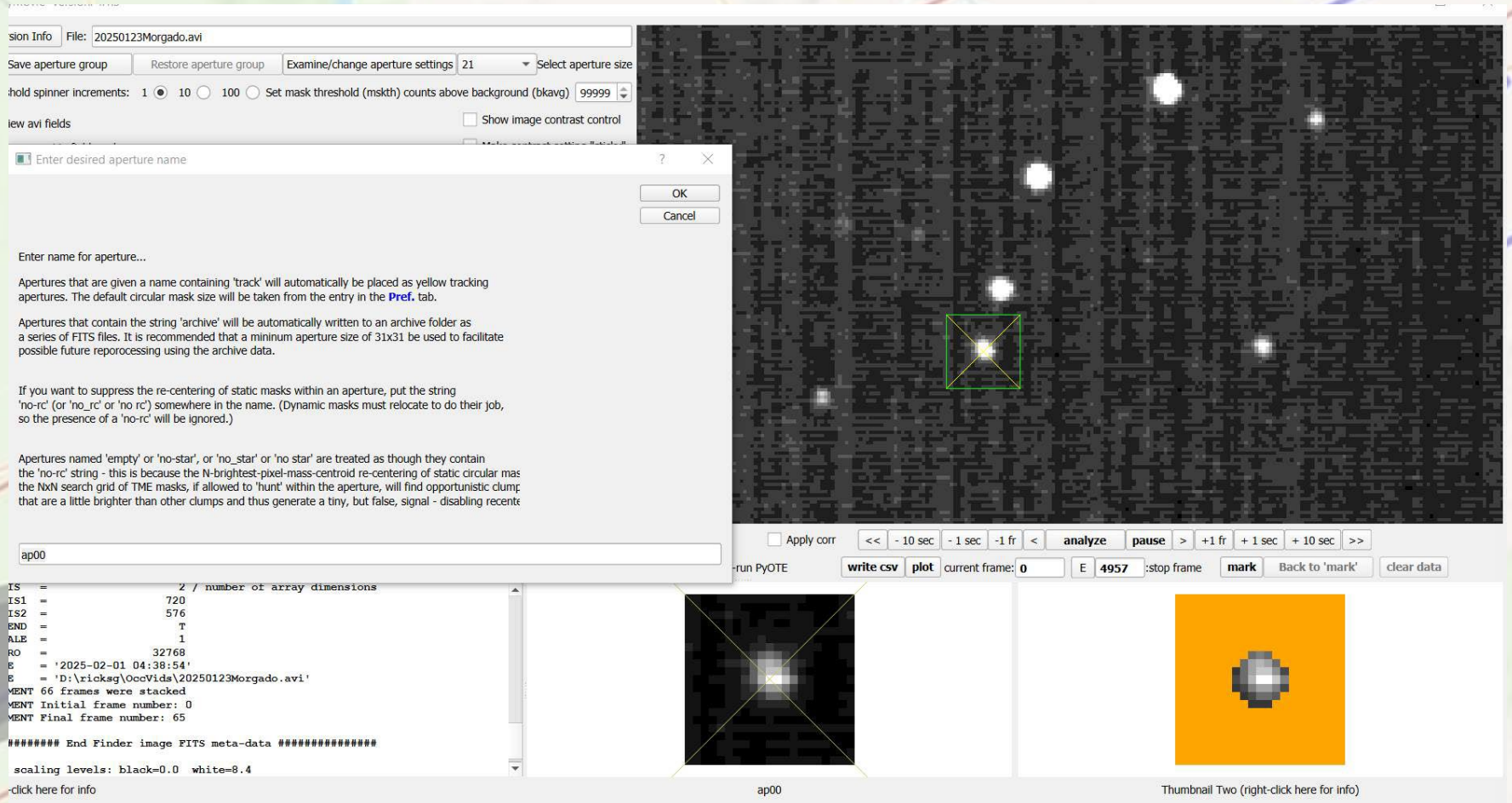
- I used to think that it was best to make your mask size big enough to ensure that every target photon was counted. I came to realize - No – the best S/N is given by minimizing the sky noise inside your mask, and that means making sure your mask nooses up tightly around your visible star image, with few dark pixels around the edges.
- For our 8SE scopes and f/6.3 or f/3.3 reducers, that typically means a mask size of 2 - 4 pixels. If seeing's bad, 3.2px or more is OK; and if the seeing is good, then usually 2.4px is optimal. You could even go about it laboriously by, in the choice box you'll see soon, doing 12 apertures around the target all at the same time, and then look at the result for each and see which one gives the best S/N.
- You should decide your *"aperture size"* first. It's the size of the box that will contain your mask and if the star wanders outside the box then the mask loses it. If it stays inside the box, it will find it. Set your box small to exclude other nearby stars. I have mine default at setting=2.1. If you change the aperture size, that aperture will disappear and you have to place it again. But for mask sizes…
- You can change your mask size even after you've set masks around all your chosen stars, by going into the *"examine mask size"* tab. But whatever numbers are set inside those boxes becomes the default used for new apertures until you change it again.

There's a wide choice on what a right-click on a star does. Here's the menu. For most well-behaved tracking situations, the best choice is *"add static (fixed circular) mask aperture"*

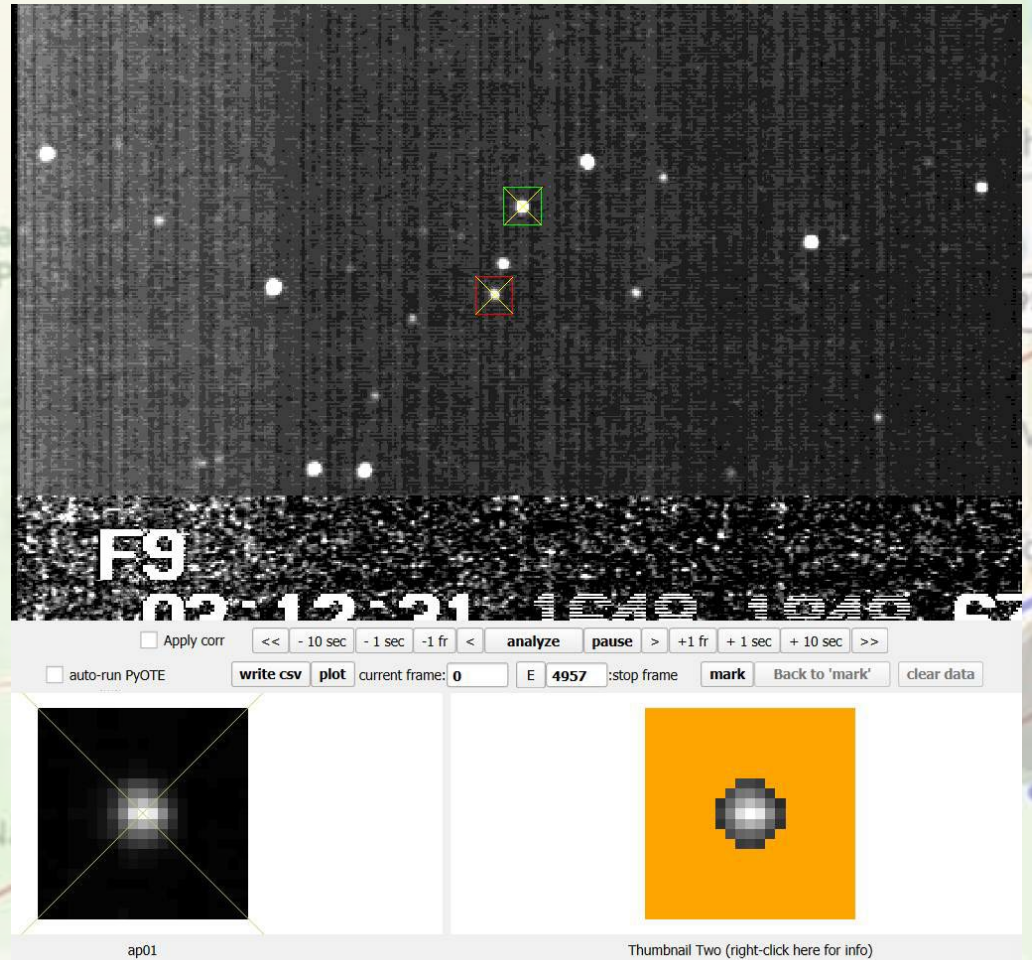


- But if the star is ***not*** well focused but otherwise no breeze or jerky tracking, then it's probably best to use a *"add TME snap-to aperture mask"*. This will find the best shaped aperture for the average shape of the target star in your finder image. You MUST first be on a "Fourier finder" if you want to use this choice. It'll warn you if you're not on a finder.
- If there's a variable breeze that jerks the image around sometimes but not always, then it's best to use a *"add dynamic aperture mask"*, which will determine the best non-circular mask for each frame individually; this will be slower processing.
- Whatever you choose, do the same for all other stars too. **For "no-star", always use a circular fixed mask**

Use your mouse wheel to zoom in on your target star, and very carefully right-click on the center brightest pixel. The box you see below will pop up and it'll ask for a name. I always call it "target" and you should too. It shows a green box around it.

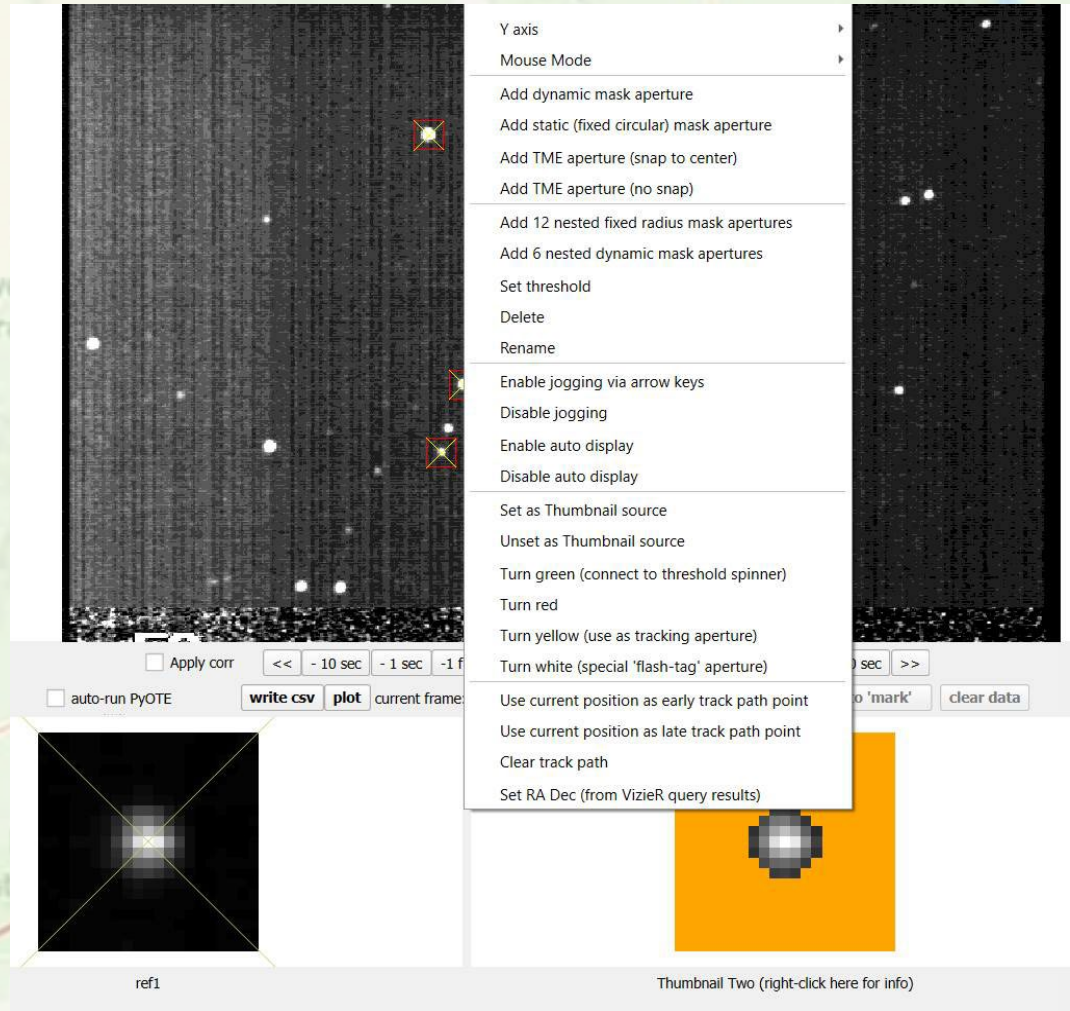


Now choose your best Reference star. This is the star which will act as the constant brightness standard for each frame. If there is variable obscuration due to clouds, having a reference star is crucial.



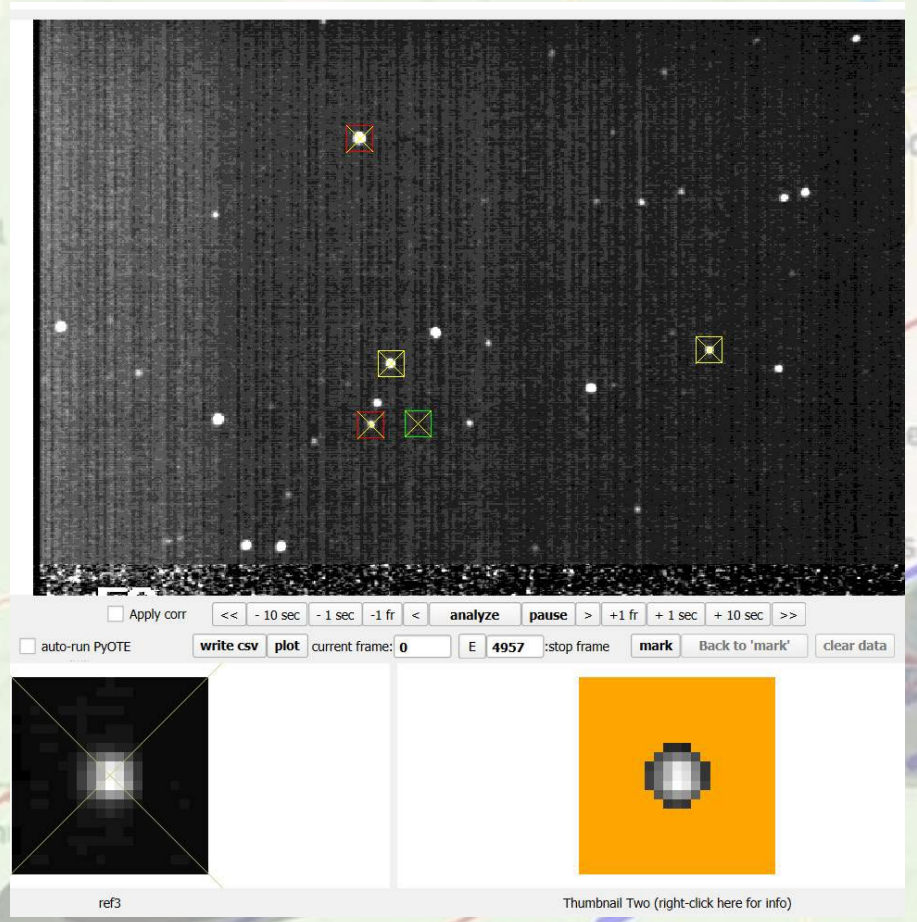
- Criteria for your best reference star:
- 1. It should be close to your target, and bright-ish
- 2. It should never contain saturated pixels, but also it can't be too faint and therefore noisy.
- Saturated pixels will show as pink or red on your orange box image (left) from frame to frame. While PyMovie is analyzing, you should hover your mouse over, and watch your prime reference star and make sure it never shows pink or red pixels on it. The saturation limit should be set a little below your bit limit. For our 8 bit chips, a little below 2**8 = 256 is 246 which is what I use. Any pixels above level=246 will be pink or red.

Two of your stars should be also function'd as tracking stars. To designate a tracking star, right click on it and click on "turn yellow…" from the pop up menu box



- **<u>How to select tracking stars:</u>**
- 1. Having 2 tracking stars guarantees that image drift ***and*** rotation will both be corrected. Especially important for targets near the zenith for our alt-az mounts, where image rotation is rapid.
- 2. The first tracking star should be as close as possible to your target, but even more important it should be bright. A saturated pixel or two is not fatal, here (but then it can't be used as a ref star).
- 3. The second selected tracking star should be far away from the first star, but make sure it doesn't drift off the chip during the analysis; that would kill that analysis and you'd have to start over.
- Ref stars can also be tracking stars, but **you can have ONLY 2 tracking stars.**

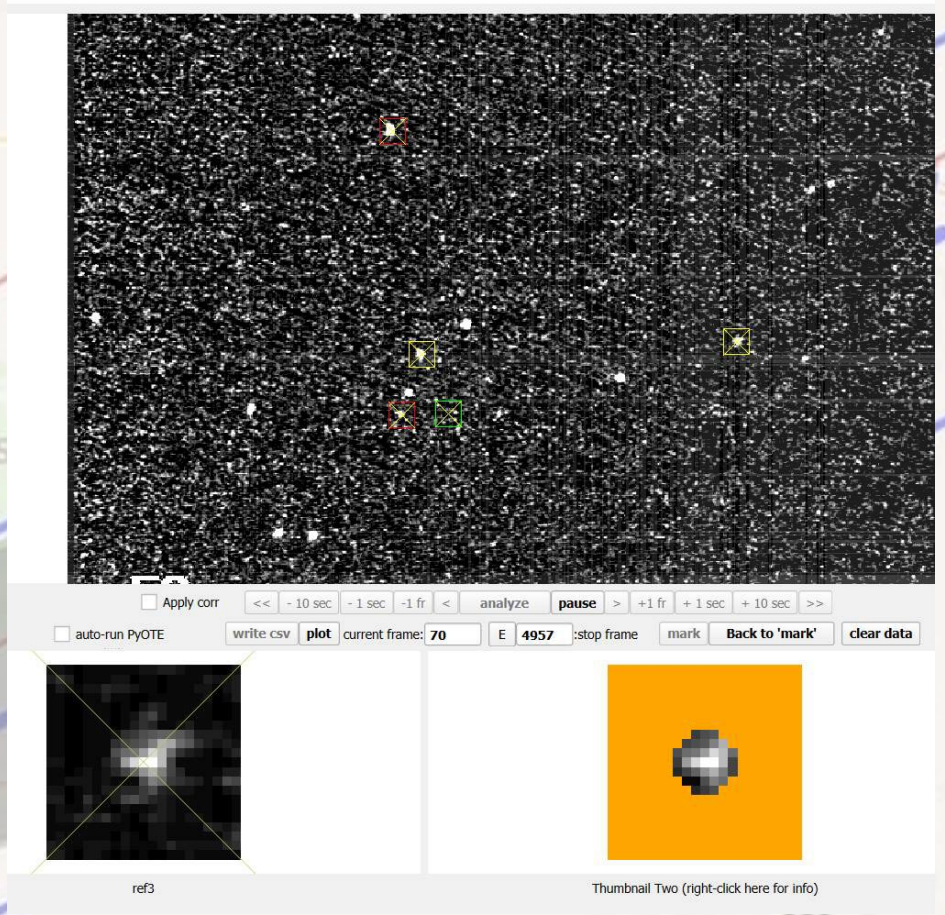
Tracking apertures turn yellow. Be sure to FIRST click and turn yellow the tracking star CLOSEST to your target. The 2nd tracker is only used to determine field rotation.



- Lastly, click on a piece of clear sky. It should look clear on your finder field and also on my finder chart.
- This aperture you need to name as “no-star”. That will guarantee that it doesn’t try to center itself on the brightest pixel, but instead blindly follows the two tracking stars.
- Your ‘target’ star and ref stars, on the other hand, should be set to center itself on the brightest pixel near the tracking location, for best data.

Now we're ready to analyze the data. When you now click on *"analyze"*, it should warn you there's no archived apertures. That's OK. You can read about them and use them if you want. I don't use them. If anyone wants to re-look at my data I'm happy to send them the full *.avi* file.

- As it analyzes, it goes frame by frame through your video and you'll see the sky background is now not silky smooth like the 'finder' stack but actual and real time sky-noisy.
- You should watch that your stars are being tracked, and you should hover over your reference star(s) for a bit and make sure they aren't showing saturated (red, pink) pixels. Also get a feel for how tight your masks are. They should be tight but not so tight you lose pixels that are mostly signal vs sky noise.
- It'll take a few minutes till its done.

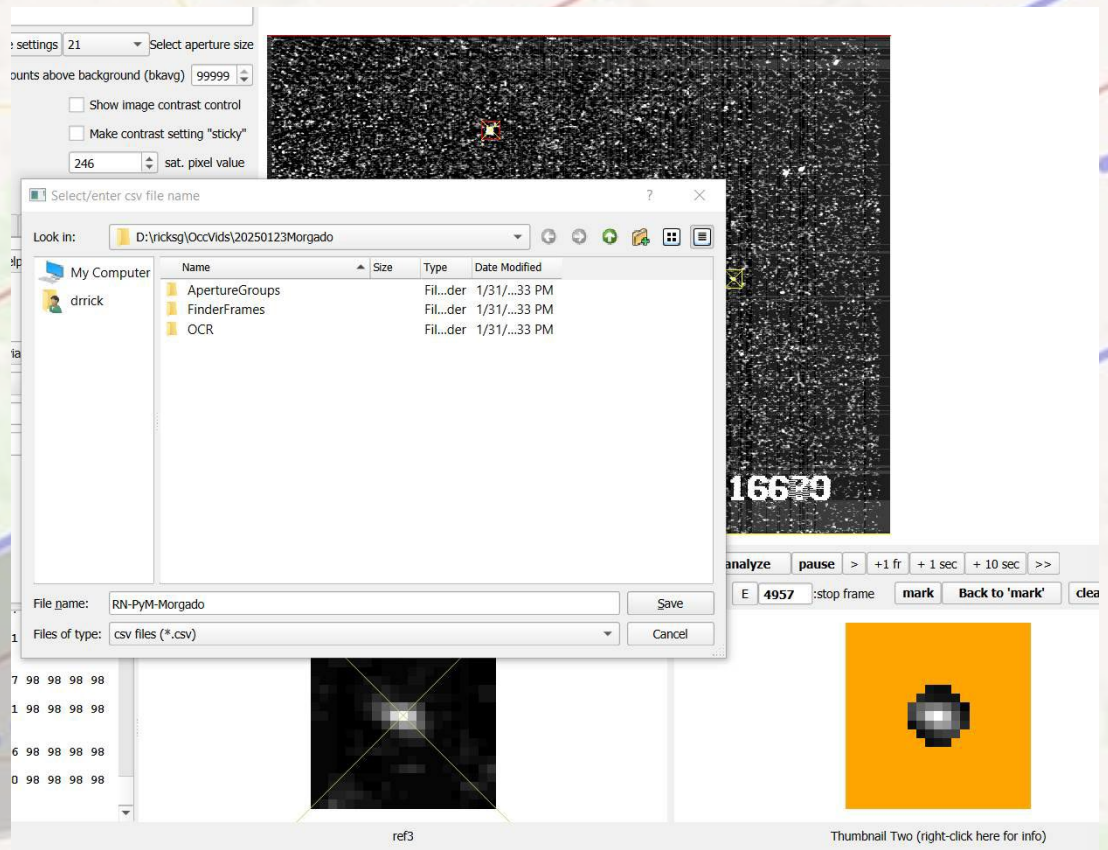


If it Looks like it Worked - First Thing to Do

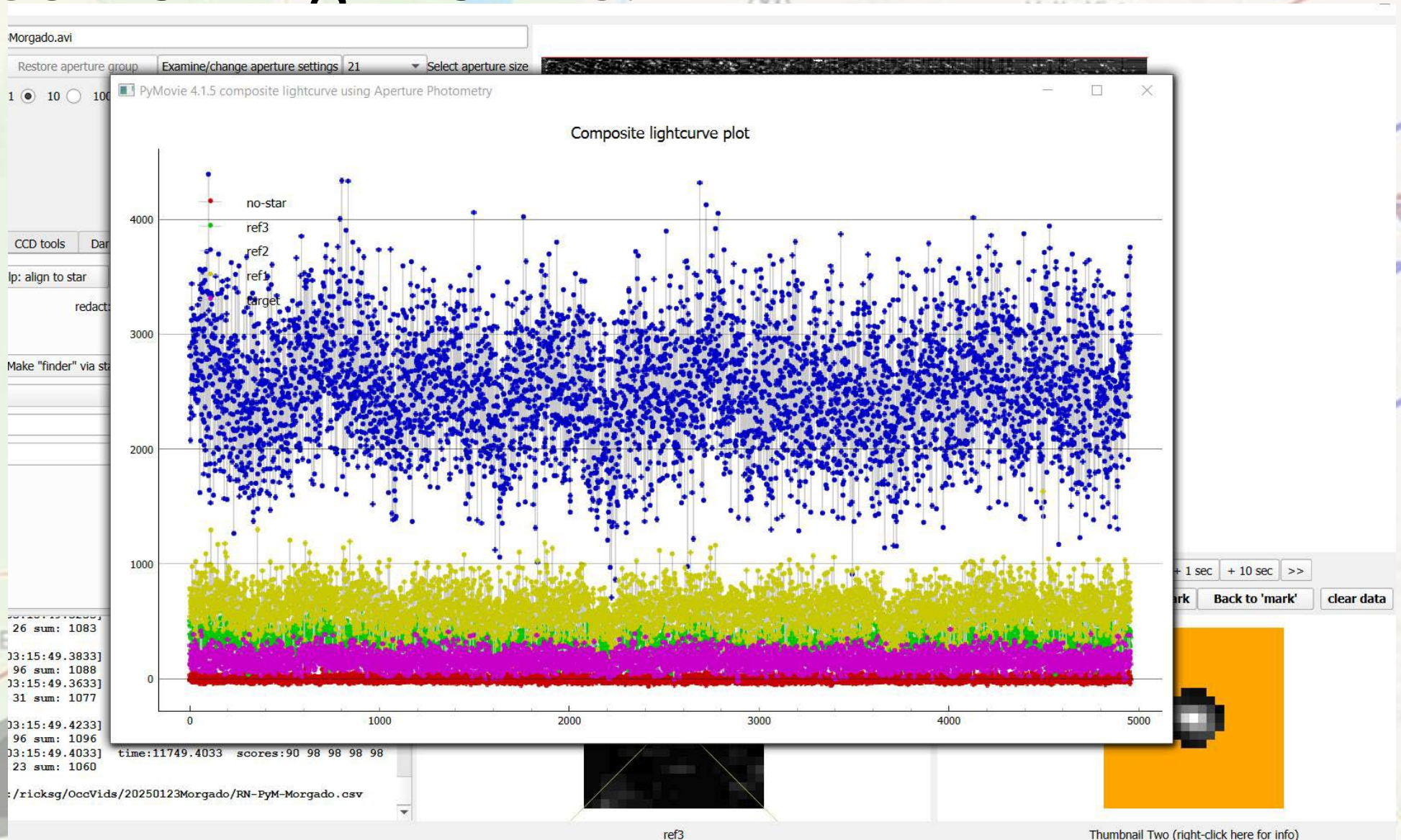
- Save the *csv* file that is the file PyOTE will use to complete the analysis of this occultation. If you forget, and bail out of PyMovie, then you have to start from scratch, which can be time consuming.
- If you choose to re-do the analysis and ignore the first attempt, that's OK, you can just use the same name for the csv file and over-write it, so nothing lost in going ahead and saving the file first thing before looking at your light curves.
- It'll automatically save it inside the folder PyMovie created to hold your other results.

When the video frames stop updating moment to moment, it's done. It doesn't signal you with a special message like "I'm Done"!

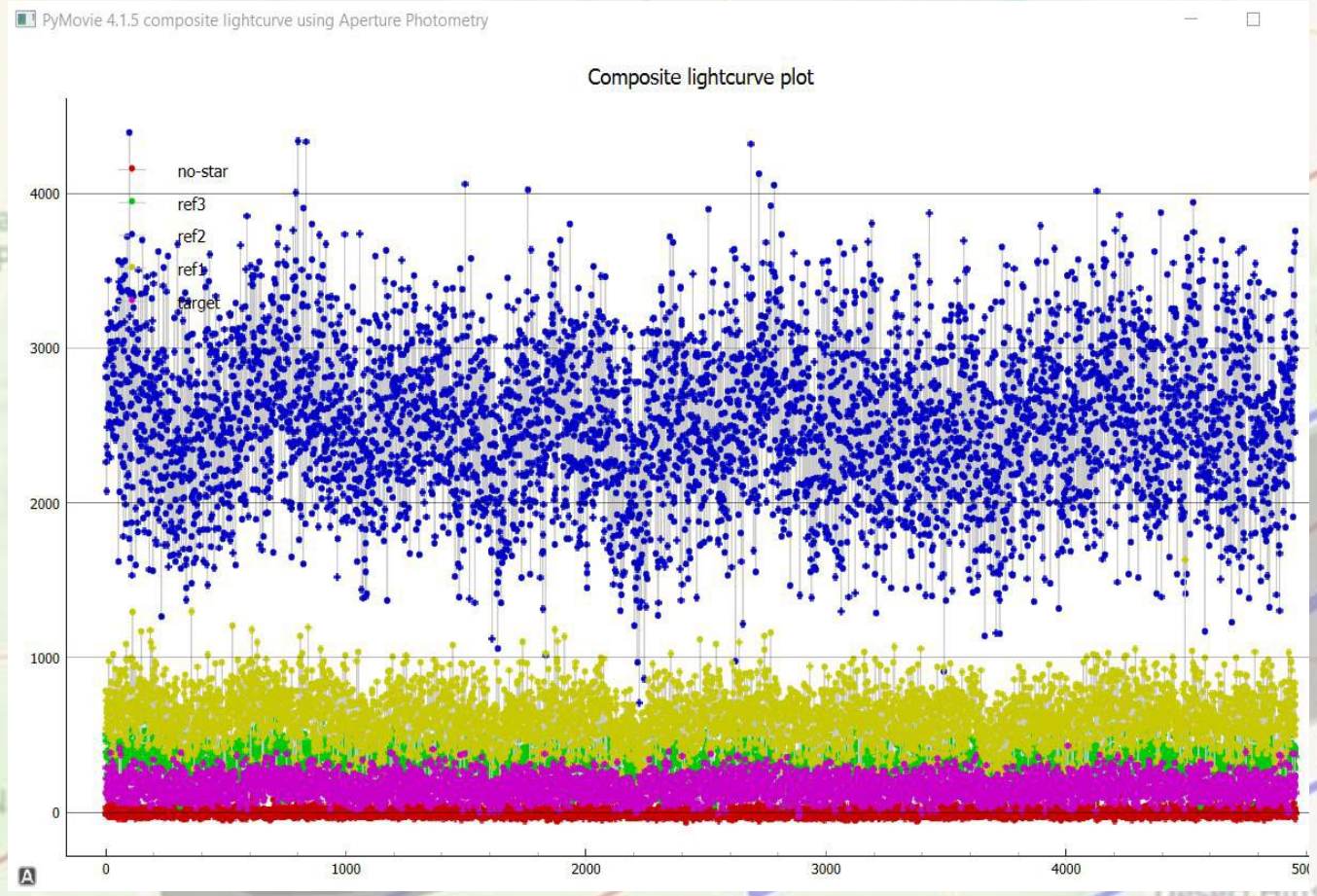
- Now's the fun part. We get to see what we got! Did we get an occultation? Or a miss? Or is it so darn noisy you can't conclude anything?
- Click on ***"plot"*** and it'll take a moment and put up several pop-up boxes on top of each other, with light curves. The top one will the the **composite** of all objects you aperture'd.
- Pause and look at it… and before you do anything else, again, save the complete photometry *csv* data by clicking ***"save as"*** on the buttons below the image. It'll pop up the folder it's made already for this event, and my convention is to name it "RN-PyM-Morgado"; my initials **RN**, produced by **PyM**ovie, and the asteroid name. Then click the *'save'* button and it'll save and thenthat box will disappear.



Click on the top-of-the-stack image, which will look something like this.

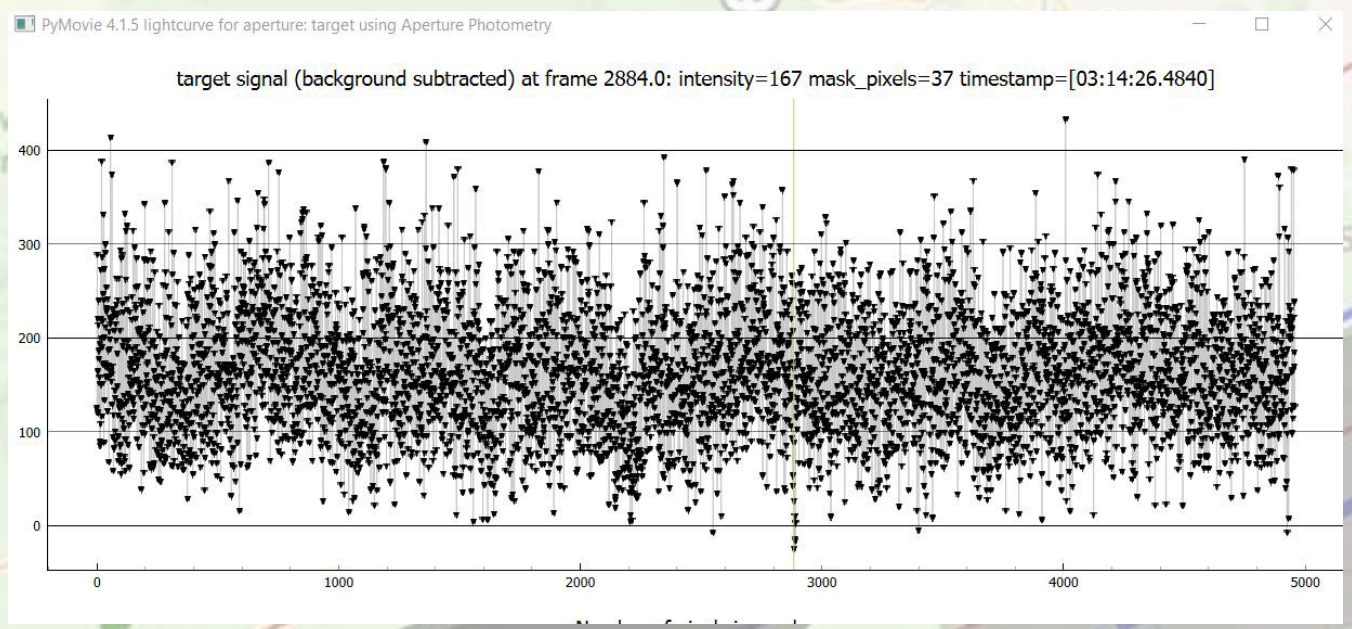


Now full sized, you can do a screen capture of this composite of light curves, and this is the first image that you send to me, for our event webpage.



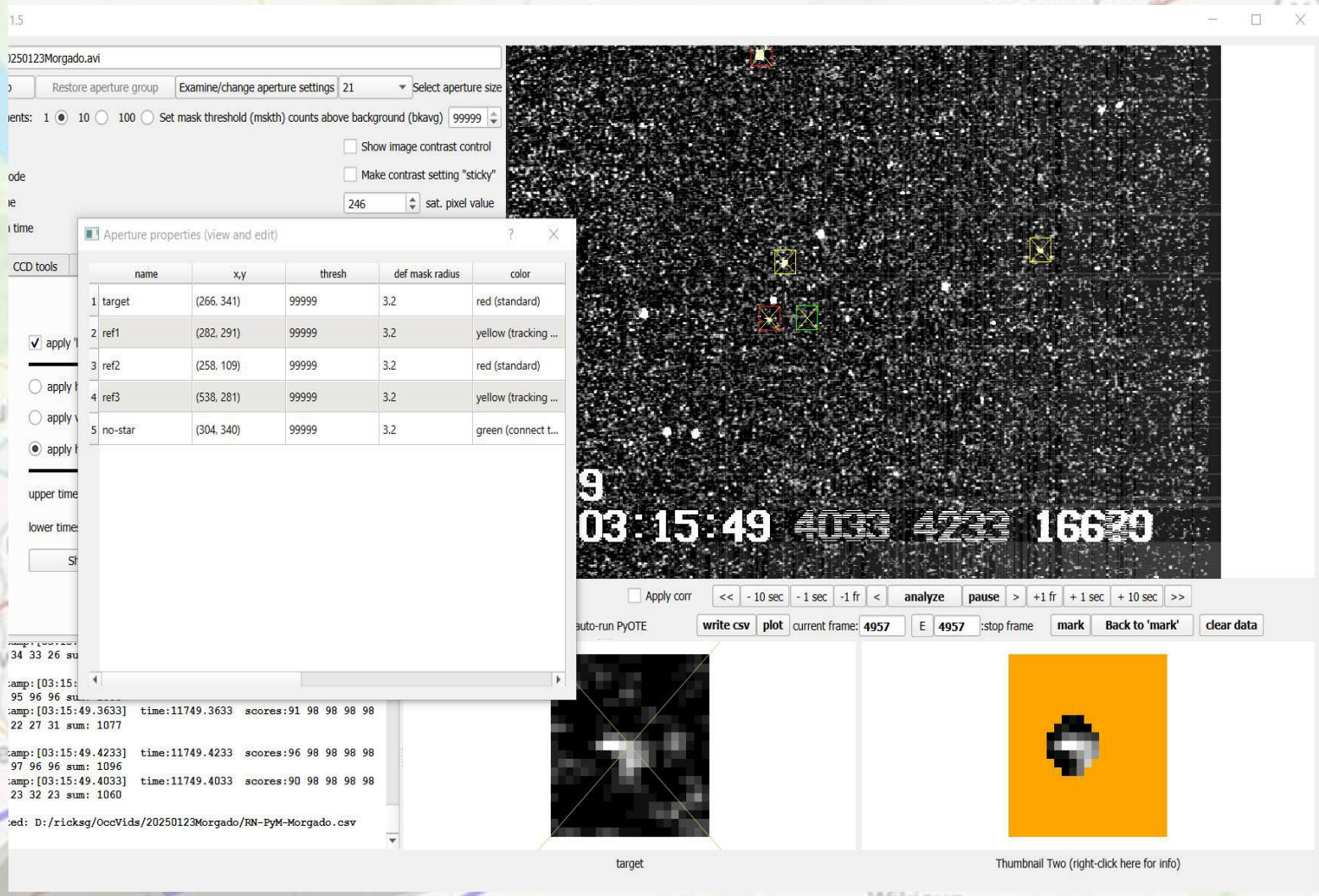
- I like using “Greenshot” a freeware screen capture program, you can specify many parameters to suit your purposes.
- But you can also, in Win10, hit “shift-SCR” keys simultaneously and that’ll capture the full screen. I can trim it down to the right size after you send it to me, if you don’t trim it yourself.

Click the ‘x’, on the composite light curve, and it’ll show the next one underneath, which should be the target light curve, it’ll look something like this… click on it.



- Now look at the finder chart for this event and remind yourself when the event time was, and then move your mouse over this image to position the vertical brown line at that event moment, as carefully as you can; the UT times are given in the title as you can see at upper right.
- While keeping the brown line from moving, then screen capture this image too, and send it to me.
- You see in the example there’s a real event, at the predicted time.

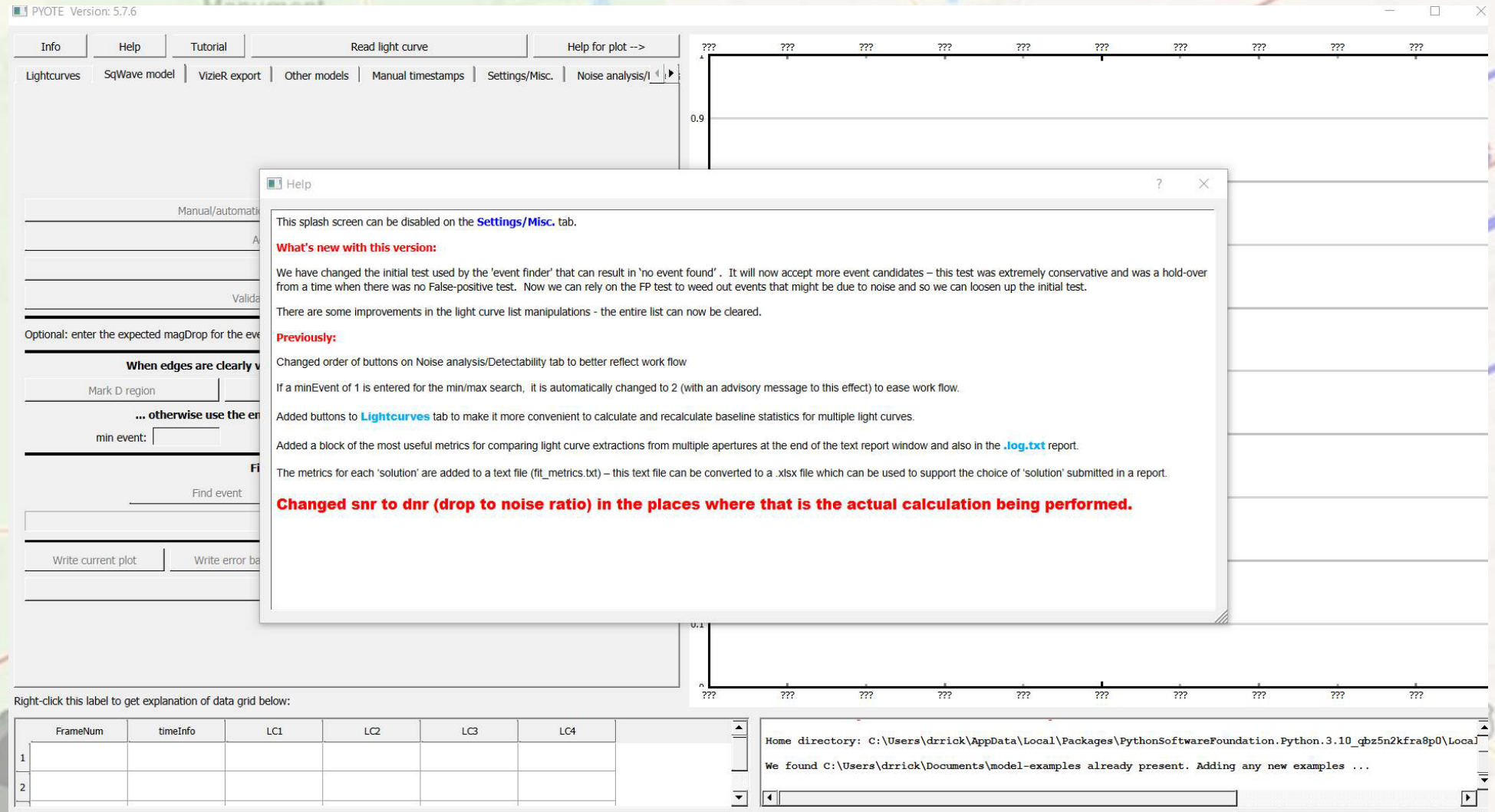
Now, a screen capture of the PyMovie screen. Get “target” to show at bottom, and click on the “median filter” tab and then click on the “examine…” button to show the apertures. Position that box exactly as you see at right. This shows me you’ve properly median-filtered, and the size of your masks. This is one of the several images you send to me for our website page.



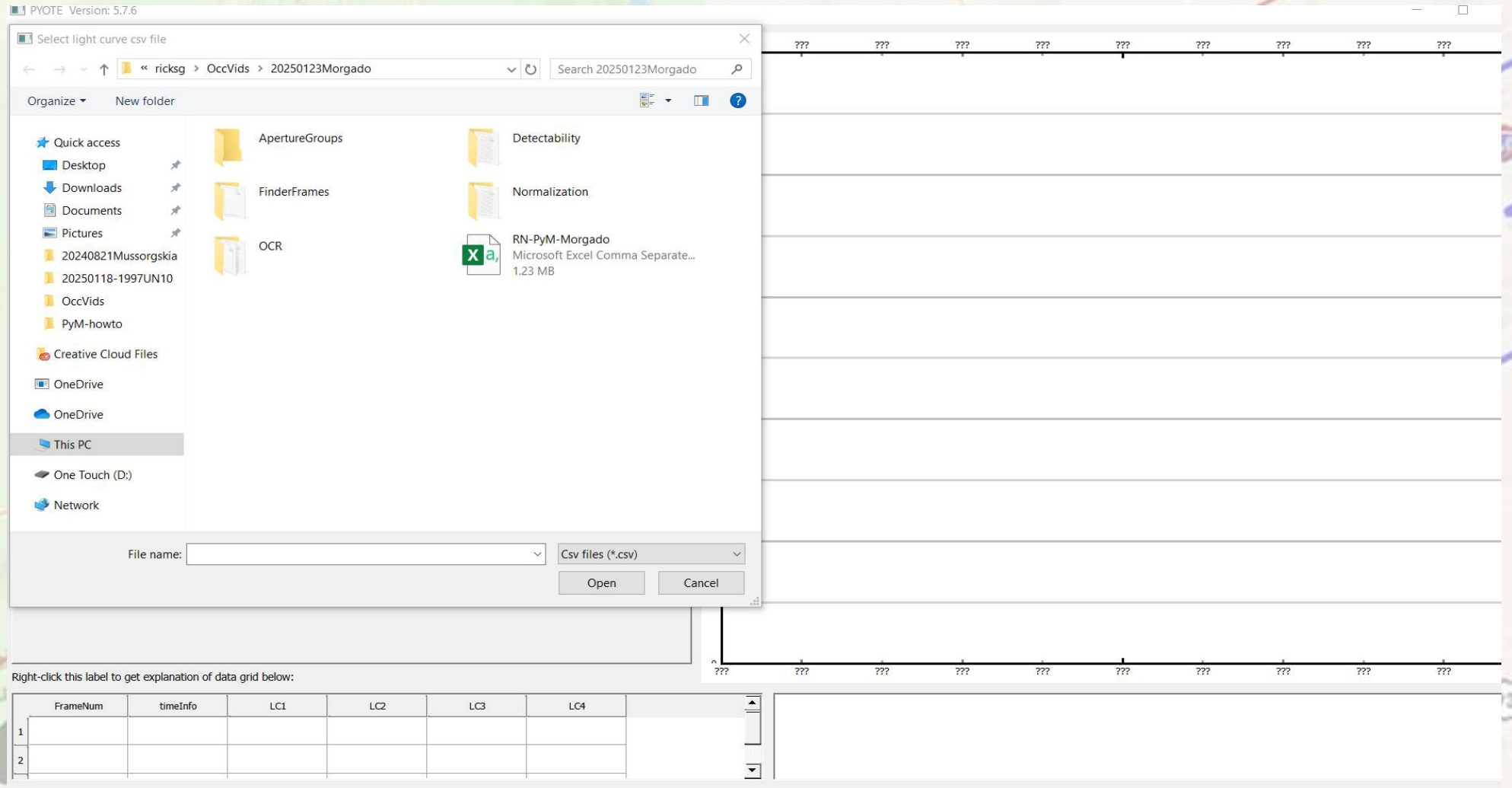
Let's Review the PyMovie Screen Captures I want to be sent to me, for our webpage for this event…

- **1.** The colorful composite single image of all aperture light curves from PyMovie
- **2.** The single image of the target's full light curve.
- **3.** Zoom in on the target light curve at the predicted time of event and screen capture that too, with the brown vertical bar at the predicted moment of central occultation
- **4.** Now, after you've removed the stack of light curves and saved your *csv* file, I want you to click on the 'median filter' tab to show what's clicked, and also on the 'examine aperture masks' tab to show what sizes are your masks, and before you screen capture, be sure it's the target star that is shown at the bottom aperture box.

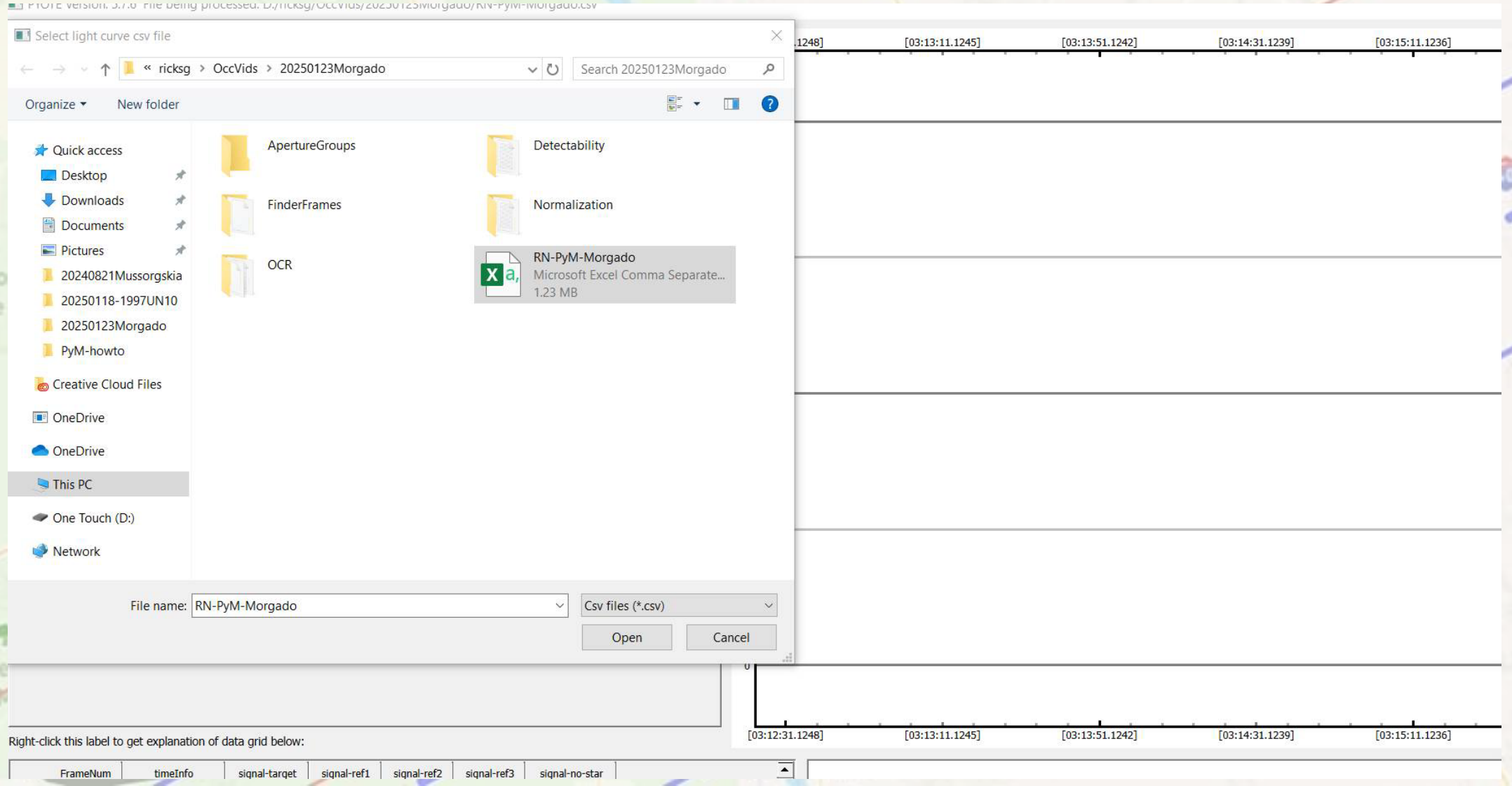
Leave PyMovie on, but now we bring up PyOTE. Go to your home screen and click on PyOTE to initiate it. It'll take a few seconds and then pop up what you see below



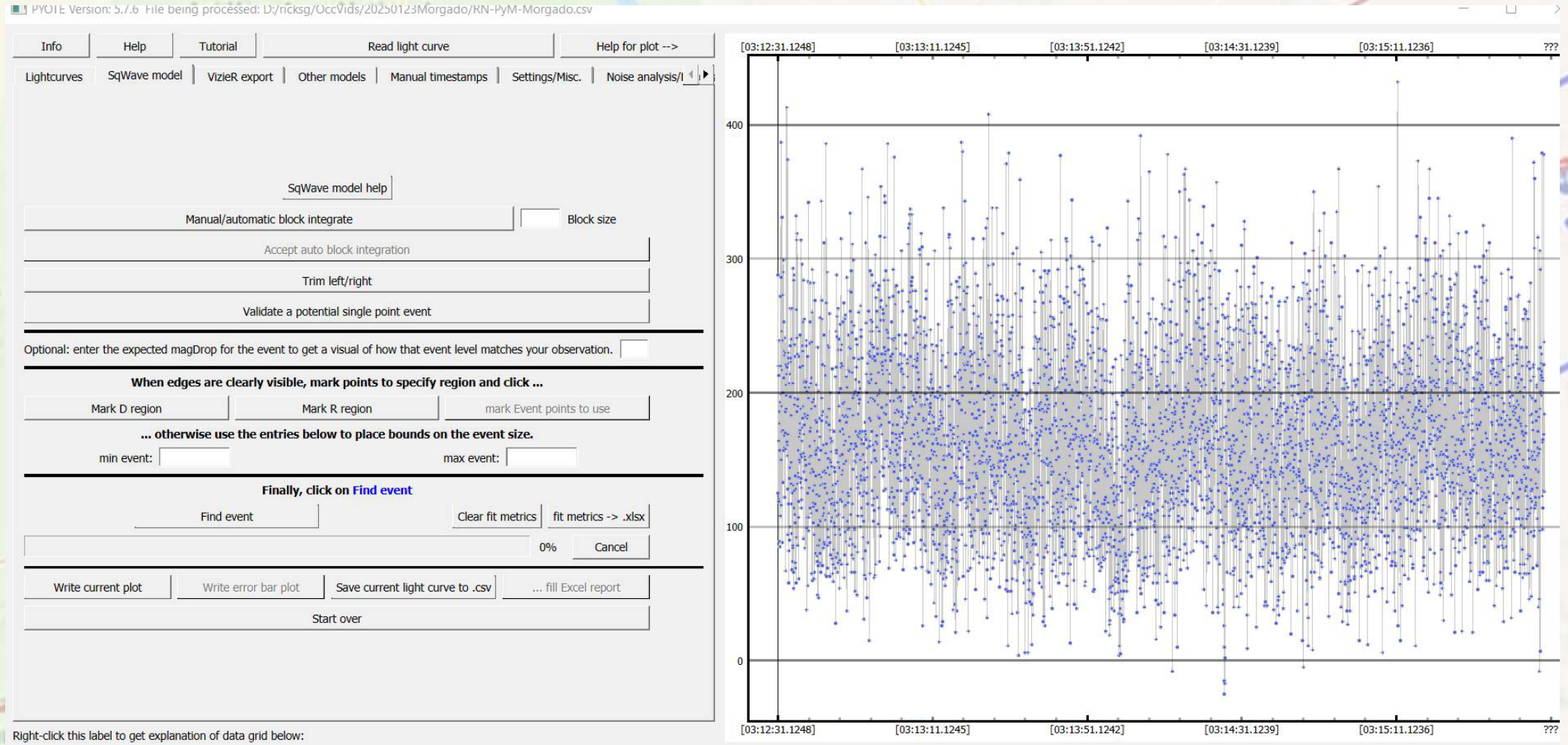
X-click the info box you saw, and you get this screen, asking you to click on your .csv file you created in PyMovie. It should automatically go to the correct folder so you can click on it



Below, I will click on “RN-PyM-Morgado” and then “open”



PyOTE will then automatically plot the first aperture, which you insured was “target” back in PyMovie.



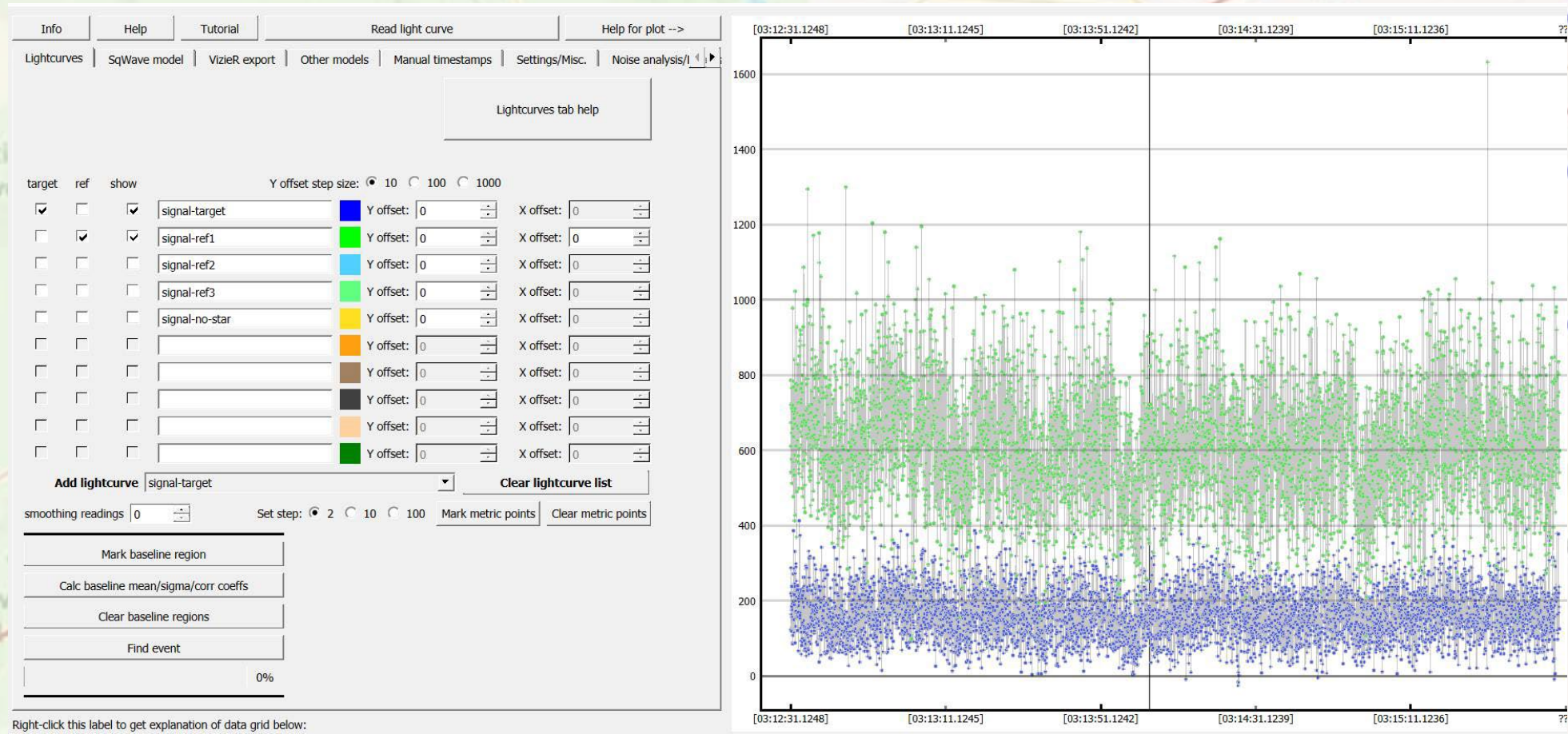
Right-click this label to get explanation of data grid below:

	FrameNum	timeInfo	signal-target	signal-ref1	signal-ref2	signal-ref3	signal-no-star
0	0.00	[03:12:31.1248]	125.000	787.000	2889.00	463.000	-7.00000
1	1.00	[03:12:31.1648]	288.000	542.000	2811.00	505.000	32.0000

```
No time step irregularities found.

timeDelta: 0.0399997 seconds per reading
timestamp error rate: 0.00%
```

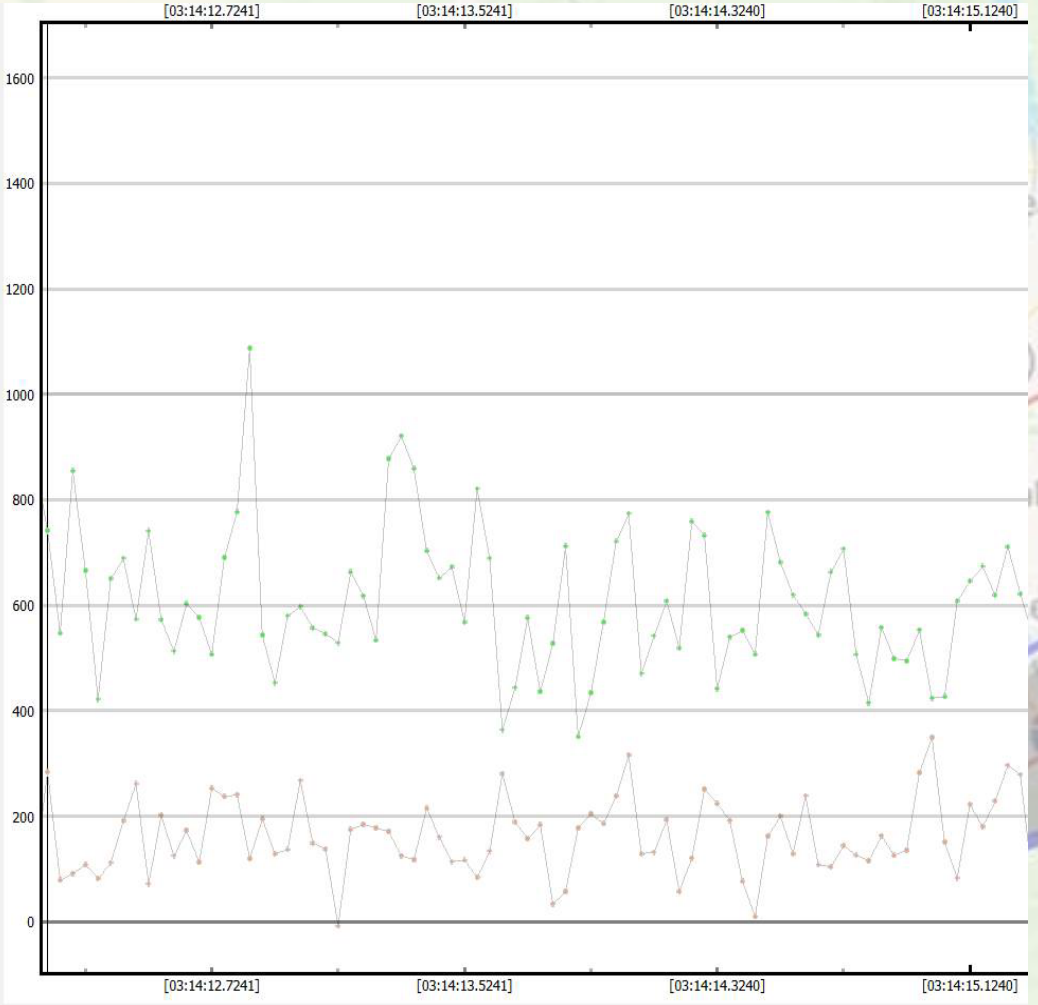
Now on the left, under *"show"* click your chosen reference star. I try to make "ref1" the name of my likely best reference star. In "ref" column, click the box for ref1. The light curve for ref1 will appear.



To Note for the Step to Follow this Slide…

- PyMovie always outputs each frame's photometry. It does not group together and output collections of frames into "integrations".
- The only exception is if you specifically check the *"analyze in field mode"* box before you start your analysis run in PyMovie. In that case, it'll output a *.csv* line for each field separately rather than combining even and odd fields into a single frame.
- Now, if you use an integration setting of, say, 16x, that means you are telling the Watec to group 16 fields (=8 frames) into a single integration before outputting it to the VTI. You might think, then, that the VTI will output only a single data line for that integration. But no, it does not. It outputs every field, even though in this case there are 8 frames (=16 fields) that are a single integration.
- Will all 8 then just be identical? Not precisely. PyMovie still outputs each frame, but it has slightly variable readout noise and so the readings will not be precisely identical within the integration. But, still you should see if you zoom in on the PyMOvie light curve, that the readings will be nearly identical in groups of 8 in my example.
- By just **looking** at your own light curves in PyMovie you should be able to tell what your integration setting was. What you will not know is the offset, for the first data actually saved on the file. PyOTE will determine that for you.

Next step; integration blocks. You should always note what your Watec setting was and mention that in your audio at the telescope. It's confirmation that PyOTE gets it correctly when it independently tries to determine it through auto-correlation calcs.



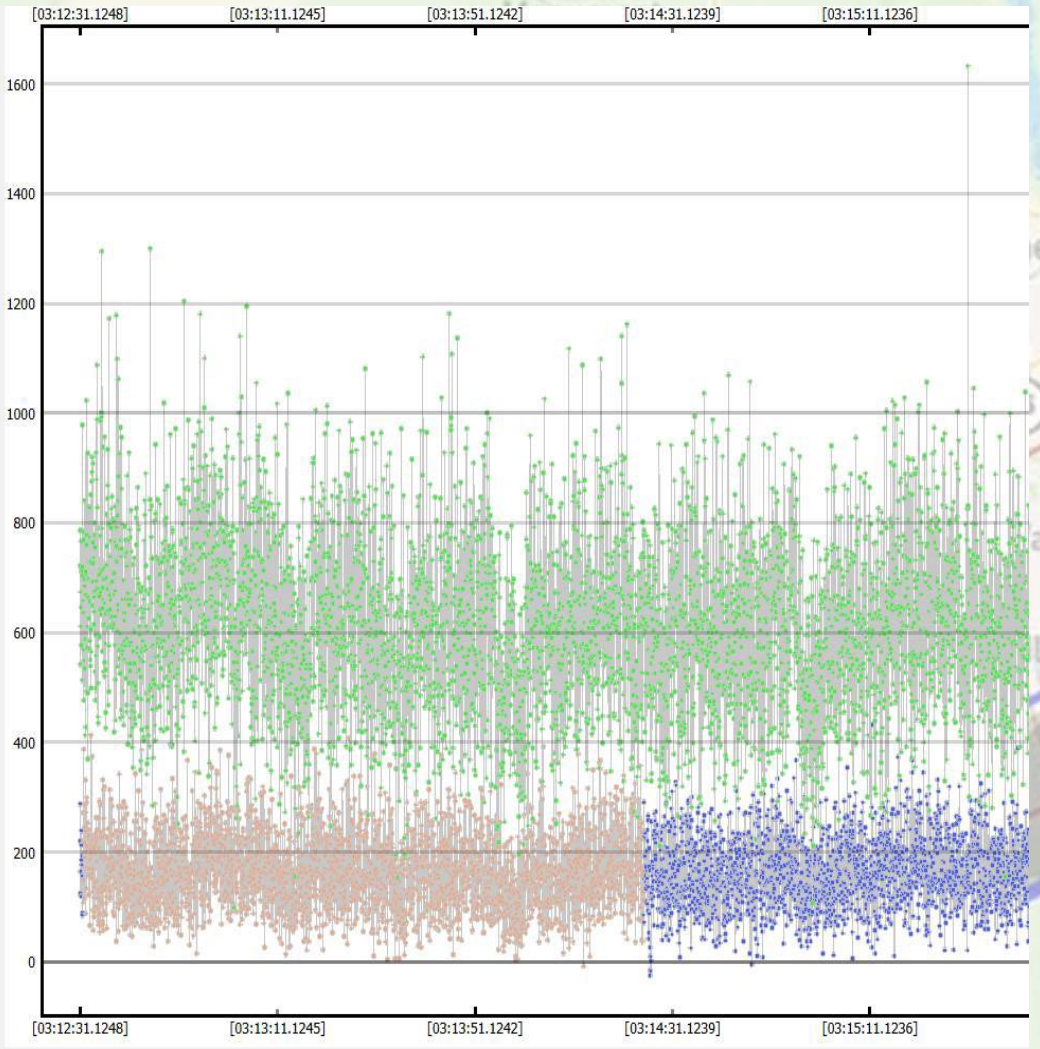
- Example: 2x means 2 fields per integration. But 2 fields = 1 frame, and **blocks are counted in frames.** So, 2x means you have no blocking. That's what I used for Morgado, and you can see in the light curve that frames are not grouped into integrations. Each frame is independent. If you used 4x setting that's 2 points (2 integrations) per block. 8x is 4 points per block, and it'd be obvious that you have 4 points with very similar level before advancing to the next 4 point block all at very similar level, etc. **If you used 2x or 1x, then skip asking Pyote to automatically determine your blocking. Otherwise, I've found it'll assume there's blocking and give its best guess, which will be wrong.**
- But if you **did** block, then click on the button "automatic block"
- Note there is no block integrations at left

If you did 4x or higher integration, then now let PyOTE find that block value **and offset** automatically, as below by clicking on the button *"automatic ...."*

- When you click the button, it'll then show progress in the right side box, and within seconds it'll find the autocorrelation analysis which determines what's the best determination of your integration setting and it's offset from the start of your data. If it finds the same setting as you know you set, then 'accept....'. And the light curve will now change, just the single points per integration.

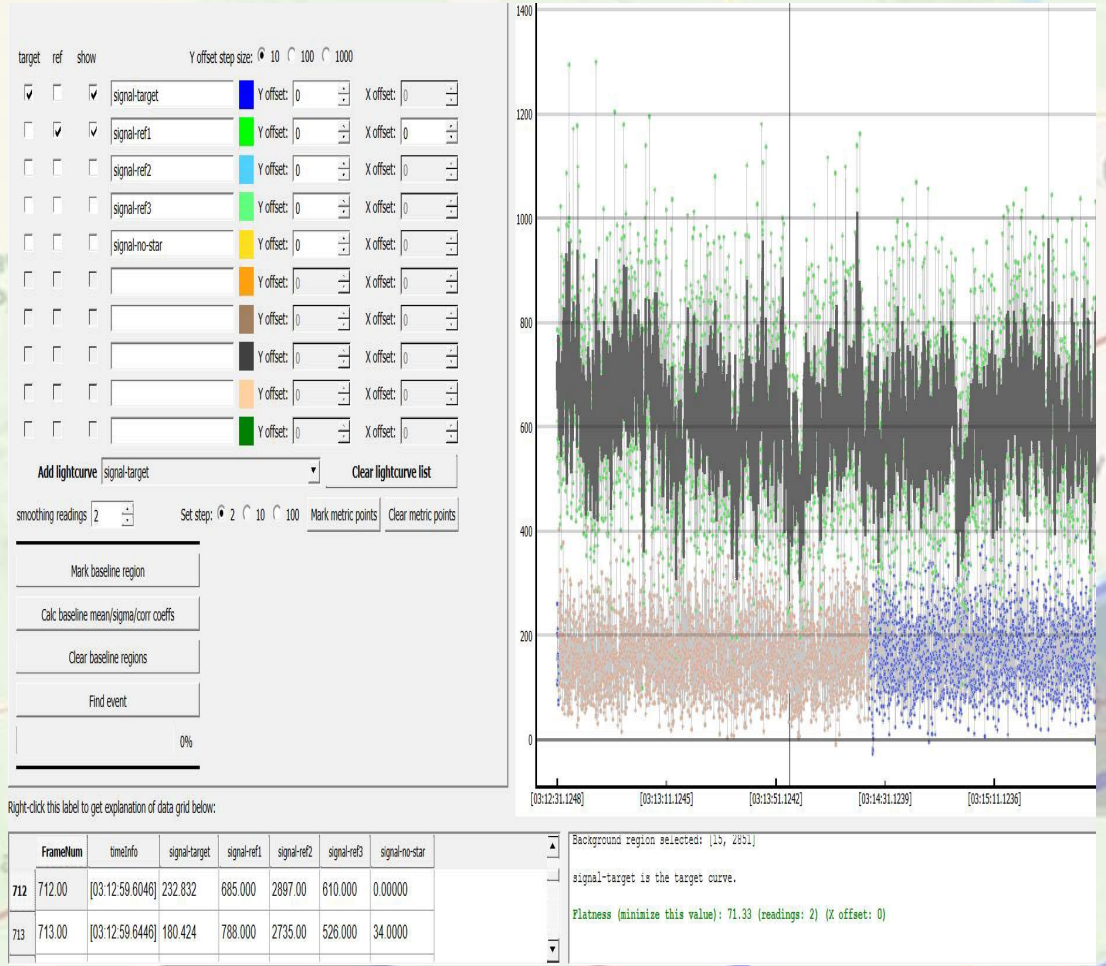
- (I need to make a screen capture of the screen showing the automatic blocking button)
- And then "accept " if it determined correctly.

Next, it's time to use your reference star to flatten the artificial wiggles in the target star's light curve



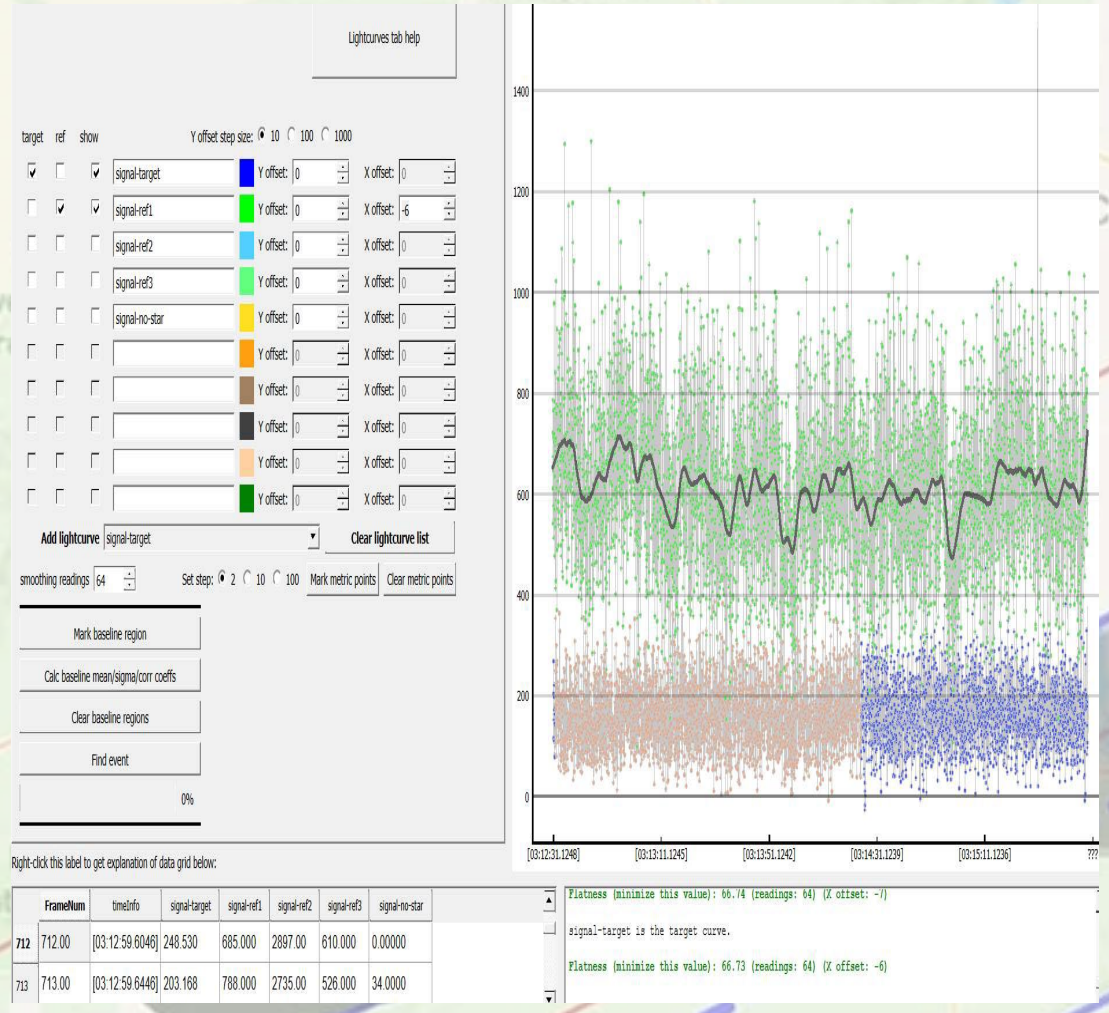
- Now, most of the rapid variations you see at left are just uncorrelated noise, not 'real'. But if you use your reference star's every brightness point, you'll be ADDING uncorrelated noise to your target – NOT what you want!
- But some of the longer period variations are real – transparency or wind generated noise that affects stars in your data. So, the art is to find how much to smooth your reference star to a curve that is the best approximation of 'real'.
- Here's how you do that…

First click on two points, highlighting them in red dots. The two points should encompass a range where you believe your target star was not occulted and should ideally be perfectly flat; your “metric interval”



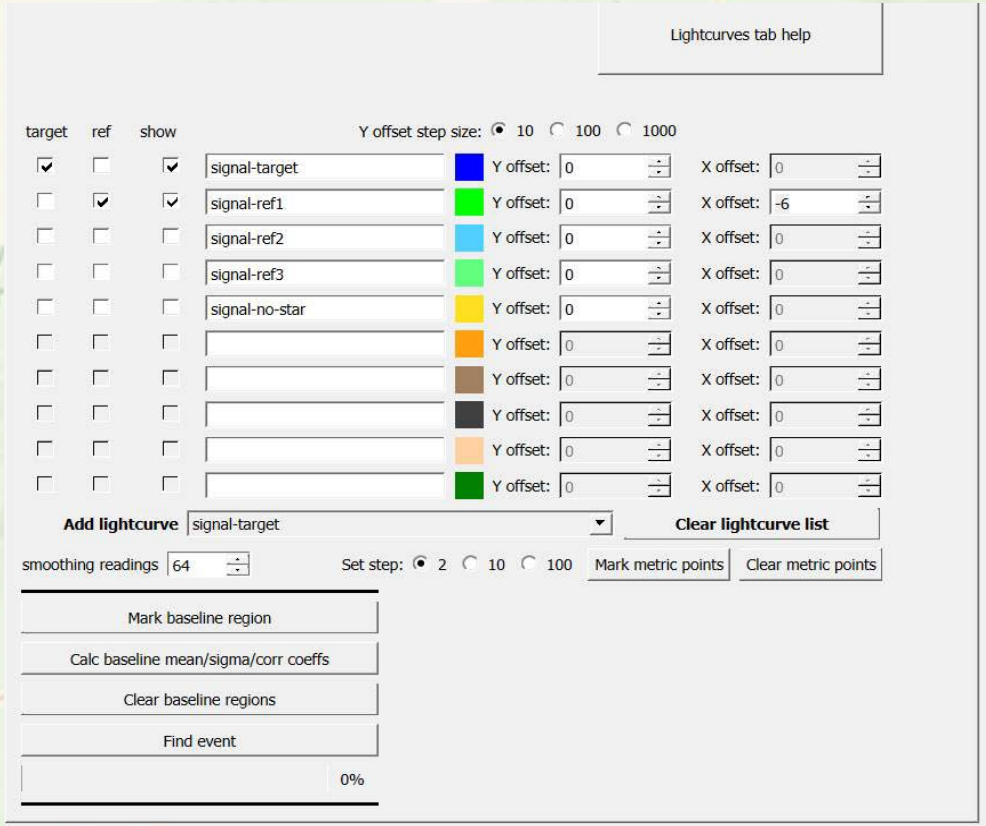
- So now click on ‘metric interval’.
- Now on the left side (middle) you’ll see arrows to increase or decrease the smoothing length.
- Click the “up” arrow and see the black smoothed curve get less wiggly and the flatness metric get smaller (in the lower right box). Keep clicking until that flatness metric stops going down and begins to go back up.

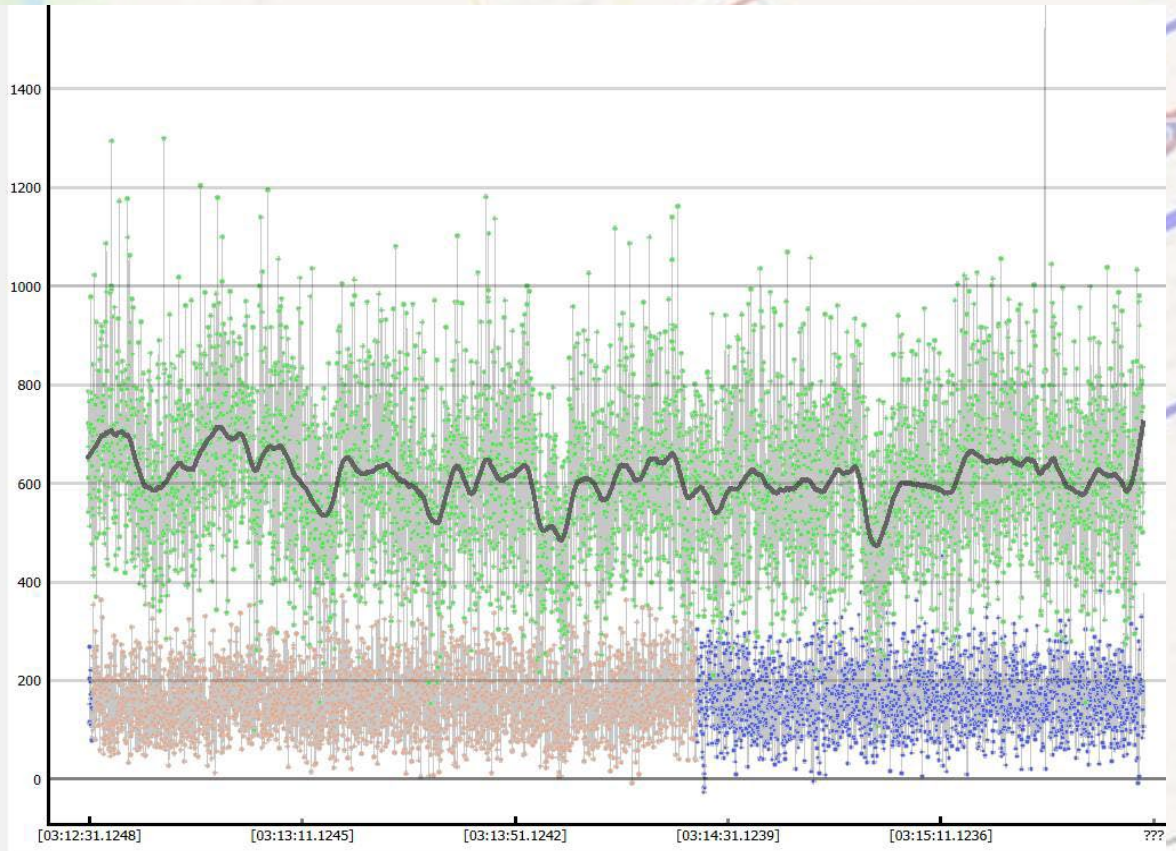
If you have drifting light cloud, you may have a “time offset” to the dips in clouded brightness. So; then adjust the number in the box to the right of your ref box, left of the light curves.



- Adjust that until the metric stops decreasing and starts going back up again. If you have no clouds at all, it may be OK to leave it at 0. It might even be best to not use a ref star at all, if even a little wiggle in the ref star is pure noise. I usually do not find that. Usually I can improve the metric by using a ref star and smoothing, sometimes though the smoothing is much more than the example at left.

You see the optimum smoothing for Morgado below. And the occultation is the narrow drop in the middle of the light curve, on the edge of my selected metric interval (PyOTE turns metric interval points to orange).





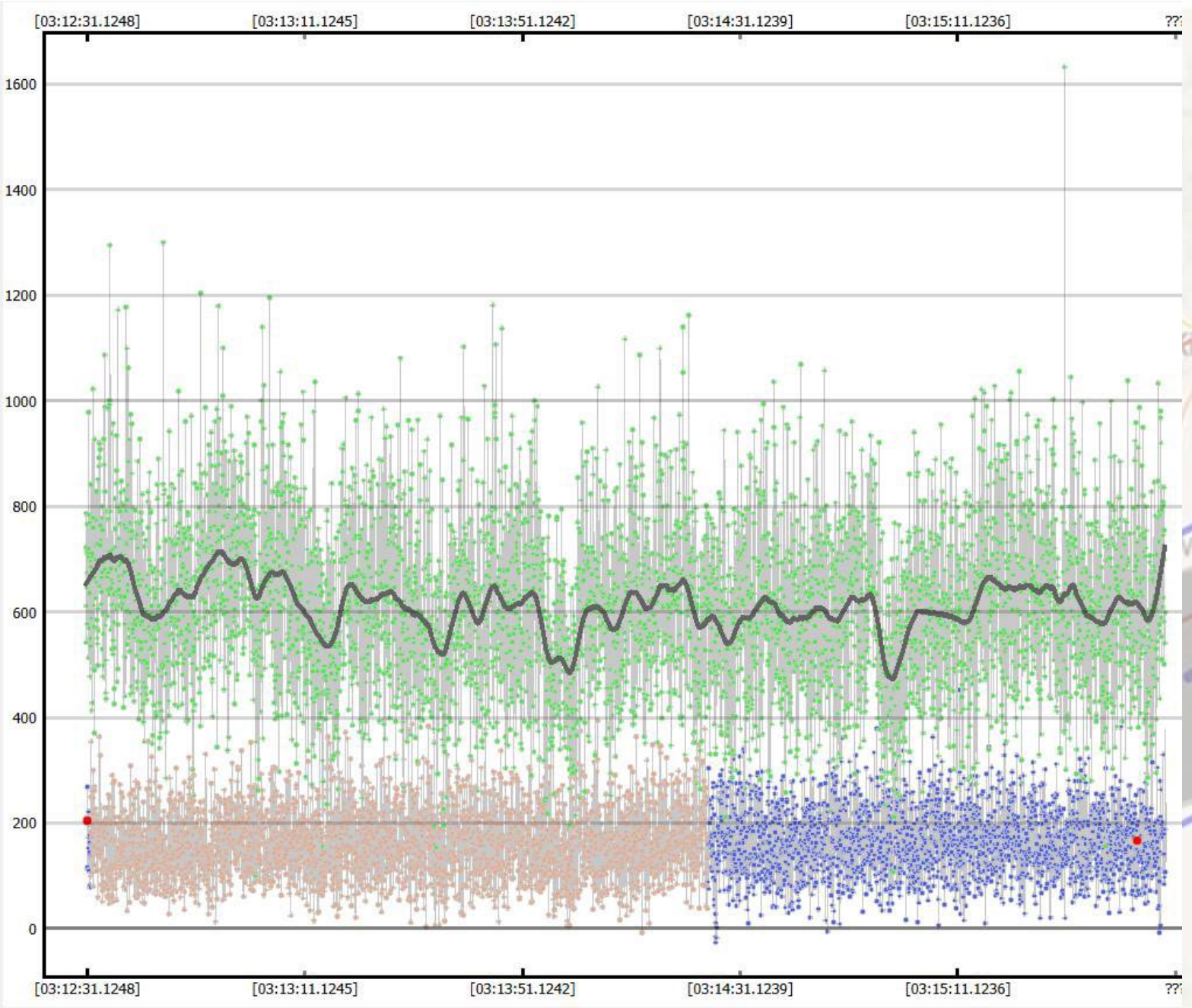
Right-click this label to get explanation of data grid below:

	FrameNum	timeInfo	signal-target	signal-ref1	signal-ref2	signal-ref3	signal-no-star
712	712.00	[03:12:59.6046]	248.530	685.000	2897.00	610.000	0.00000
713	713.00	[03:12:59.6446]	203.168	788.000	2735.00	526.000	34.0000

```
Flatness (minimize this value): 66.74 (readings: 64) (X offset: -7)

signal-target is the target curve.

Flatness (minimize this value): 66.73 (readings: 64) (X offset: -6)
```

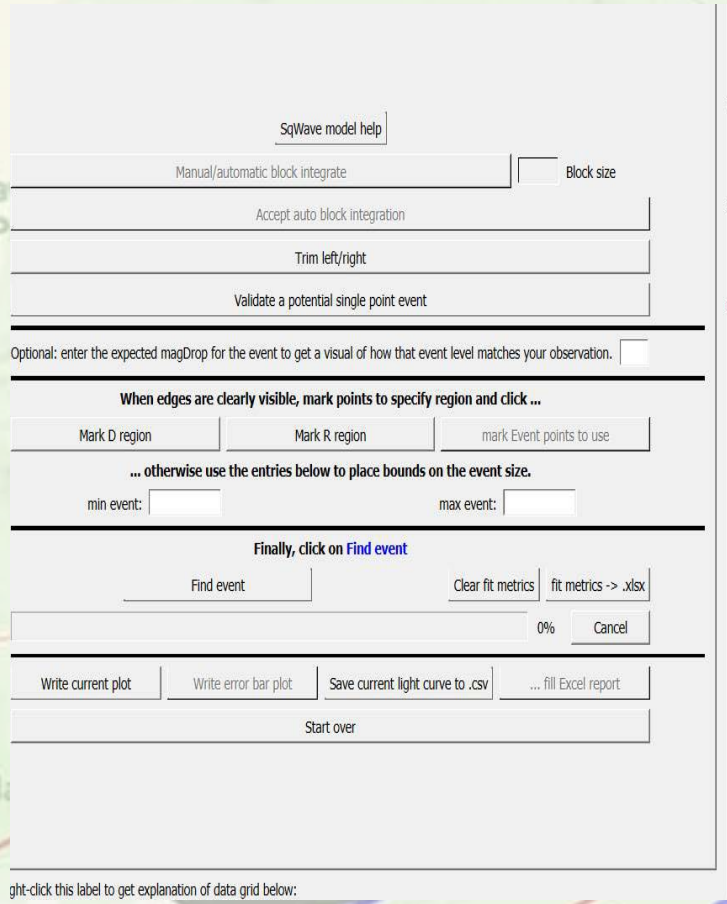


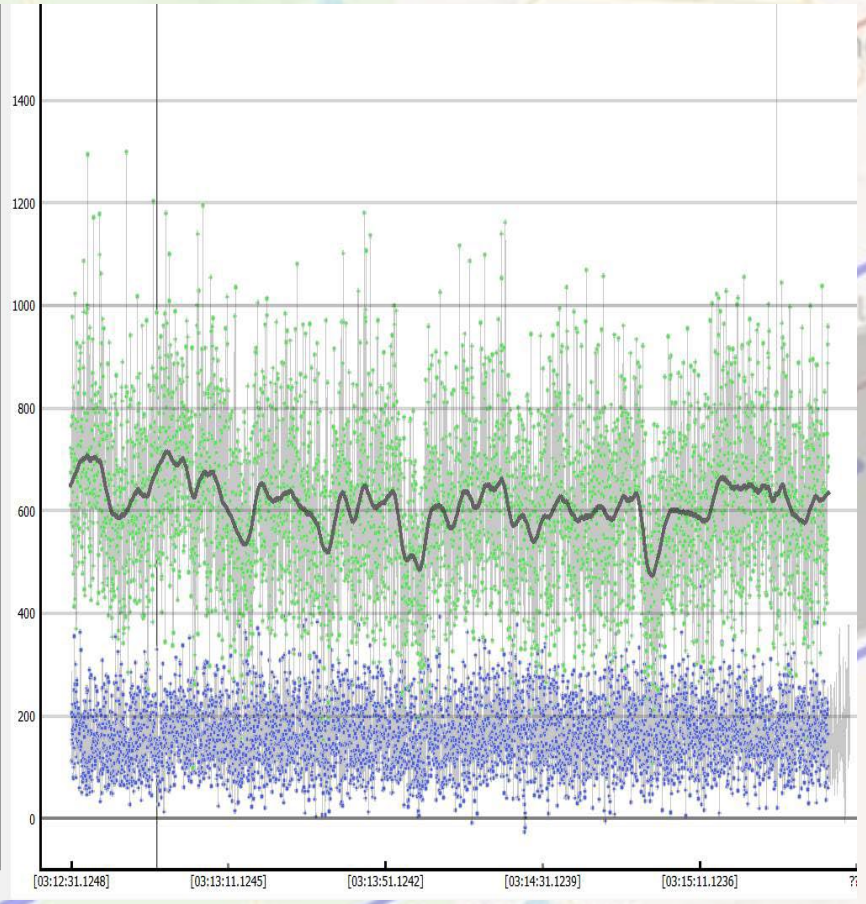
Screen Grab and crop the image of this light curve with metric interval and smoothed black ref star curve.

I just want just the light curve, not the entire PyOTE screen.

This you will mail to me.

Next step is to trim out any bad data on the edges. If there's bad data at the begin or end, or even interior, then click on two light curve points that enclose just the data you trust and of course the event too.





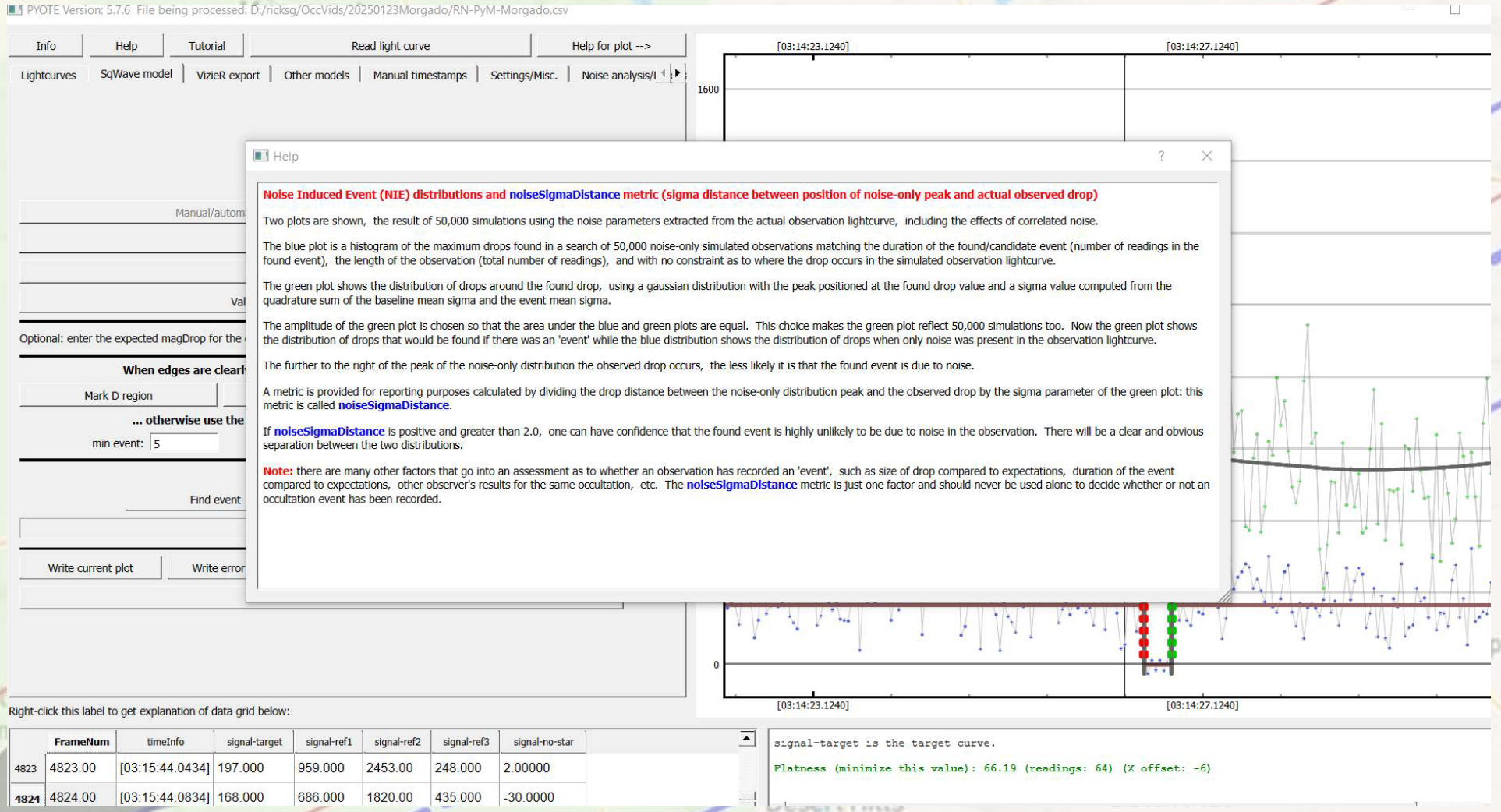
- Don't be too miserly. Keep the data unless you really see it's contaminated bad..
- Then click "trim" button and the bad data outside will turn gray.

Now, you've got an optimally smoothed reference star, you've trimmed out bad data. You're now ready to find the timings of your D and R.

- Click the *"square wave"* tab to find the simplest method of determining your D and R timings. If the occultation is sharp and obvious, you can click on two points that enclose the "D", then click "D interval" and then two points that include the "R" and click "R interval", and then click "find event"
- A more sophisticated analysis might be called for if the data is unusually good on a bright star and you see evidence of Fresnel Diffraction. Not covered in this intro of mine.

- The alternate way, especially if it's a very short event, is to pick the number of points that the event is definitely SHORTER than your occultation, and another number that is definitely LONGER than your occultation, and then click "find event".
- Doing it this way, I find I have to click on "find event" 3 times before it actually completes.

After it's done, PyOTE will look something like this. You can see the red and green D and R and accuracies underneath the info advice screen



This PDF not done yet… more to come when I have time.

- PyOTE will pop up images now, including the full light curve with timings
- …. And the ‘false positive test’ results. For the latest version of PyOTE the ‘false positive test’ is two humps for the distribution of values. If these two humps center offset from each other by more than 2.0 standard deviations, and there’s no reason to think that’s for some artificial bad situation (car headlights, passing through phone wires, bad pixel cluster, dust mote…) then I report it as a ‘positive’, with the timings.
- You must send me the ‘false positive test’ graph as part of our website page.
- Also, the full PyOTE light curve with red/green timings, and also the zoom in on the event itself. These should have first had you position the vertical brown line on the predicted time of the event, as shown on my planning page.