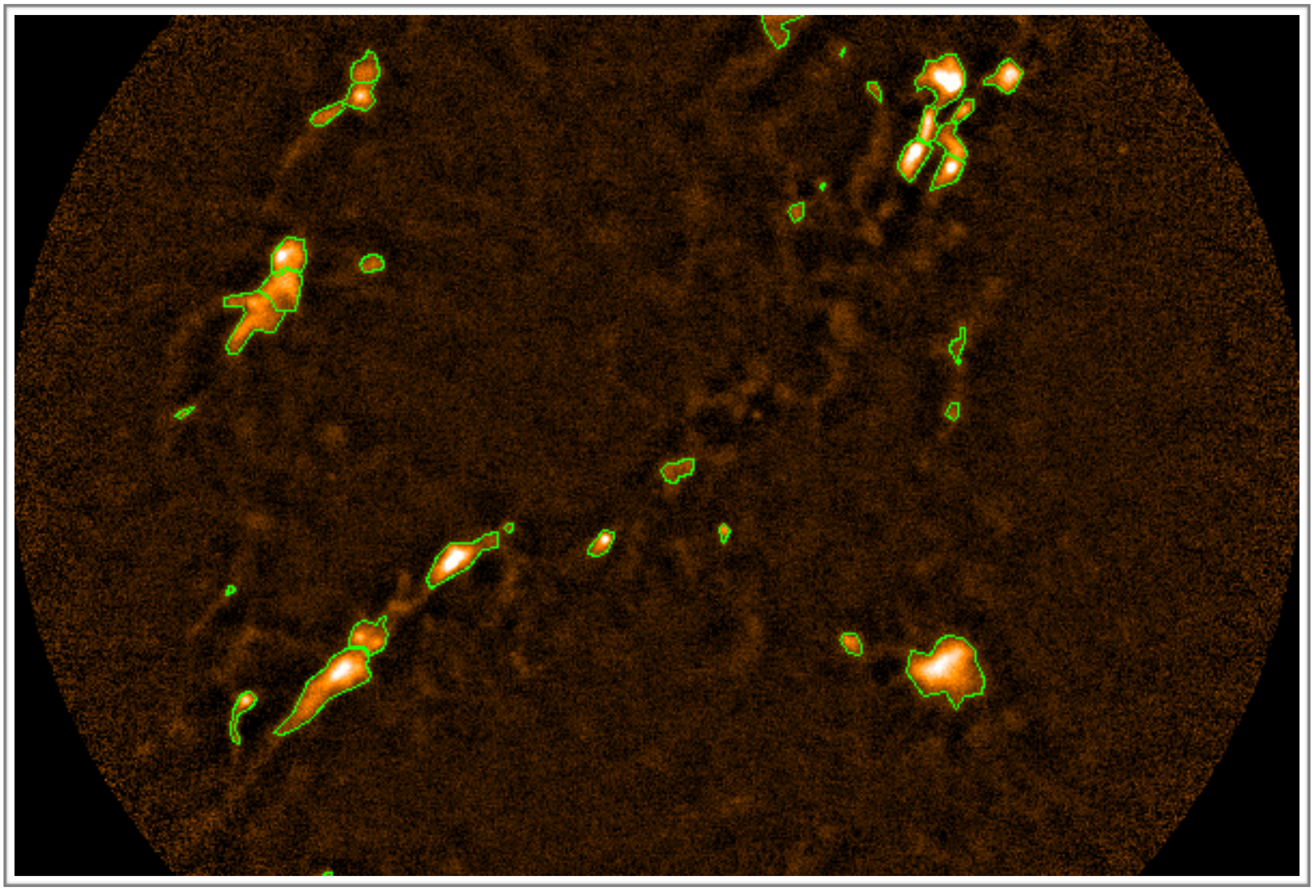
Assignment 2

Clumpfinding and Synergy of Data Sets



ASTR 351L

Spring 2019

Finding Collapsing Cores

And Using Multiple Datasets

Link to Introductory Slides:
https://www.eaobservatory.org/~s.mairs/ASTR351/slides/ASTR351L_CF.pdf

Procedure:

Part I

1. Download the file (21 MB):

   https://www.eaobservatory.org/~s.mairs/ASTR351/assignments/Clumpfinding.tar.gz

   This is a compressed file containing a directory called "data" that is home to a SCUBA-2 (850 micron, continuum) map of a small piece of the southern extension of the famous Orion Nebula, a large star-forming region in the direction of Orion (Kaheiheionakeiki). In addition, it has a folder called "config" (hosting 1 file that will be useful for identifying significant areas of 850 micron emission), a folder called "FW_results" (empty), and a folder called "YSO" (containing a catalogue of YSOs).

2. Copy this file to a suitable directory to work in (e.g. your local Desktop directory, *~/Desktop* , in the following example):

   ```
   %cp Clumpfinding.tar.gz ~/Desktop
   ```

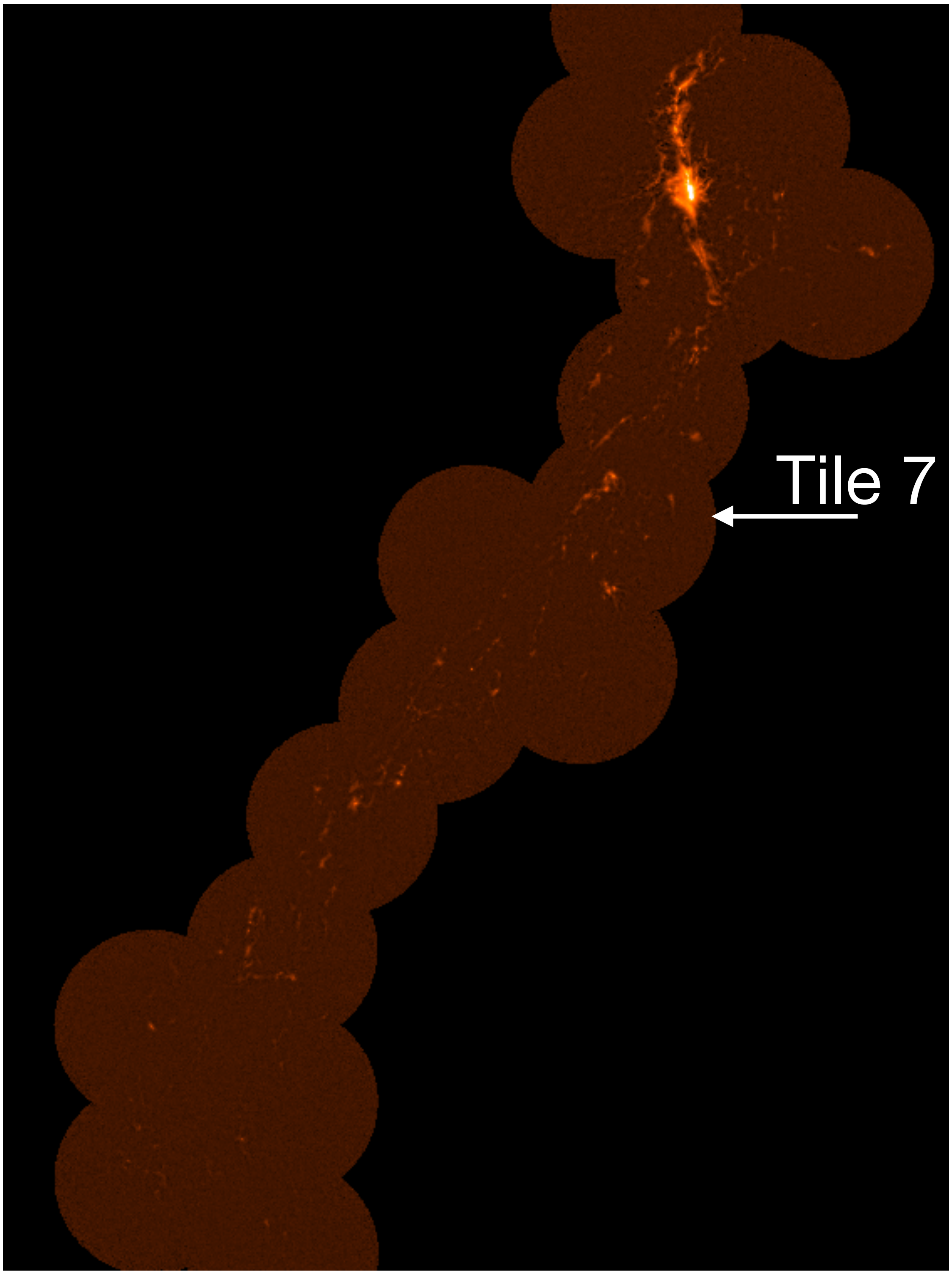
3. Switch to the working directory using "cd", unzip and untar the file using "tar -xvfz" and change to the newly-created directory containing the data:

   ```
   %cd ~/Desktop
   %tar -xvf Clumpfinding.tar.gz
   %cd Clumpfinding/data
   ```

   This directory contains one *.sdf* file called "jcmts20141026_00044_850_TILE7.sdf".

This can be checked by listing the contents of the directory by typing: %ls

jcmts20141026_00044_850_TILE7.sdf is a pong observation taken on 2014-10-26; it was the 44th scan of the night. The “TILE7” refers to its place in a larger map of the whole Orion Nebula:



4. Load the Starlink Software:

   ```
   %export STARLINK_DIR=/opt/star-2018A
   %source $STARLINK_DIR/etc/profile
   ```

   Then, take a look at the file with Gaia:

   ```
   %gaia jcmts20141026_00044_850_TILE7.sdf
   ```

   How about those noisy edges?! That doesn't look quite as nice as the map on the previous page, right? We'll fix that in the next step. Take a moment to play around with Gaia to get familiar with the region of sky - note the regions that you would consider significant just using your eyes. Remember, bright regions = emission from clumped together, star forming dust!

5. *Depending on the scope of a project, astronomers sometimes only need 1 image like this to perform all their necessary science - but in many cases, they need to use tens, hundreds, or even thousands of images like this to perform proper statistical tests on large numbers of star-forming cores. Recognising significant emission by eye is good - but the last thing you want to do is individually analyse each bright region yourself when you have hundreds of images. And what if you make a mistake and have to redo everything? The best thing to do is to script some kind of automated procedure that will allow your computer to do in seconds what it would take you weeks to do manually. Then, if the results seem funny, you can just re-run the script with new parameters. To demonstrate this, we are going to run a clumpfinding algorithm to identify emission that our eyes would pick out. It's an art, really! Some people spend weeks or months tweaking parameters to recover emission that is robust (no fluffy, noisy regions or cosmic ray spikes, for instance).*

   To help our clumpfinding algorithm avoid noisy areas of the map that it may think is "real emission", we are going to crop out the fuzzy edges of our map, where there is less total exposure time than across the smooth map centre.

   With your map open in Gaia, click on "View" (drop-down menu) —>
   "Select FITS HDU/NDF". In the box that pops up, click ".MORE.SMURF.EXP_TIME" then "Data". Then, change the "stretch" of the map so "low" is set to 0 and "high" is set to 15. This way, you can see the extent of the map that has uniform coverage, and the edges that have much less uniform coverage.

Measure the radius of the region that appears smooth (bright white). You can do this by right clicking and dragging the mouse over the image. The units change from arcseconds, to arcminutes:arcseconds.

Next, if you are into running python - refer to the Jupyter Notebook that can be downloaded here:

https://www.eaobservatory.org/~s.mairs/ASTR351/assignments/assign2_JN.tar.gz

And continue on to step 6. If you don't like python and would prefer to run the Starlink commands straight from the terminal, read the instructions below.

— — — — — — — — — — — — —

NON-PYTHON INSTRUCTIONS:

Starlink has a built-in cropping program ("CROP_SCUBA2_IMAGES") that takes a file of input parameters and crops your map accordingly. The file of input parameters is a simple text file - we'll call ours "crop_params.ini". So, open a new text file by typing:

```
%gedit crop_params.ini
```

this will bring up a word processor that you can type in and save to your current directory. In the text file, write:

```
[CROP_SCUBA2_IMAGES]
CROP_METHOD = CIRCLE
MAP_RADIUS = ?????
```

But, instead of "?????", you'll put in the radius of the region that has a consistent exposure time **IN ARCSECONDS** (which you measured earlier).
Save the file.

Then, to run the cropping program, simply type (in the terminal):

```
%picard CROP_SCUBA2_IMAGES -log f -recpars crop_params.ini
   jcmts20141026_00044_850_TILE7.sdf
```

"picard" is the name of the package that includes the program "CROP_SCUBA2_IMAGES". "-log f" means "save the output log to a file".

“recpars” stands for “recipe parameters”. Once run successfully, there will be an output file called “jcmts20141026_00044_850_TILE7_crop.sdf”. Open that with Gaia to see the difference!

————————

6. From here on, we are going to used the cropped version of the data. There are a multitude of clumpfinding algorithms we could use to identify emission structures. A shortlist would be: “FellWalker, Astrodendro, ClumpFind, GaussClumps, Cutex, GetSources….” Etc etc. All of these algorithms have their own pros and cons. We will be using FellWalker - an algorithm designed and coded by Dr. David Berry (full description: https://arxiv.org/pdf/1411.6267.pdf - also see the slides from class). Starlink can run FellWalker using the program “FINDCLUMPS”, which is part of the package “cupid”. So, we first need to load the package by typing:

   %cupid

   Included in the “Clumpfinding” folder you downloaded, there is a directory called “config”. Just like with the cropping program “CROP_SCUBA2_IMAGES”, there is a program called “FINDCLUMPS” that takes input parameters in the form of a text file. In our case, that text file is called “FellWalker_config.lis” and it has already been prepared for you. Honing these parameters can be a lot of work - and your results will change depending on what you pick! FINDCLUMPS includes many configurable parameters. If the parameter is not found in the input text file, its value is assumed to be the default, pre-programmed value. For more information, see:

   http://starlink.eao.hawaii.edu/docs/sun255.htx/sun255ss5.html.

   Take a look at the contents of “FellWalker_config.lis” by typing:

   %cat ~/Desktop/Clumpfinding/config/FellWalker_config.lis

   Or by opening the text file using the word processor: “gedit”

   %gedit ~/Desktop/Clumpfinding/config/FellWalker_config.lis

   Here is a quick run-through of the important parameters in FellWalker_config.lis:

AllowEdge: If set to a zero value, then clumps are rejected if they touch any edge of the data array. If non-zero, then such clumps are retained

FwhmBeam: The FWHM of the instrument beam, in pixels. Sources smaller than a beam will be discarded. I have set this to "0" so the algorithm might find sources slightly smaller than the nominal telescope beam…but I have also set the parameter "MinPix" to help throw away obvious noise spikes. See below.

MaxBad: The maximum fraction of pixels in a clump that are allowed to be adjacent to a bad pixel

Noise: Defines the data value below which pixels are considered to be in the noise. No walk will start from a pixel with data value less than this value.

FlatSlope: Any initial section of a clump (as the program attempts to climb towards the clump's peak) which has an average gradient (measured over 4 steps) less than this value will not be included in the clump.

MinHeight: If the peak value in a clump is less than this value then the clump is not included in the returned list of clumps.

MinDip: If the dip between two adjacent peaks is less than this value, then the peaks are considered to be part of the same clump.

MinPix: The lowest number of pixel which a clump can contain.

MaxJump: Defines the extent of the neighbourhood about a local maximum which is checked for higher pixel values.

There are still more parameters to consider, but we are effectively trying to teach a computer how to find significant emission: i.e. what is real, and what to ignore. Our eyes are generally very good at this naturally, but computers need to be taught!

So, let's run FellWalker on our Cropped data! The only piece of information you'll need is the background RMS of the data file. So, go ahead and **measure the background noise (RMS) before you continue.**

Navigate to the top level of the Clumpfinding directory:

```
%cd ~/Desktop/Clumpfinding
```

and run the following command (all on one line - remember to replace the RMS value with the actual value you measure in the map!).

```
%findclumps in='data/jcmts20141026_00044_850_TILE7_crop.sdf'
config='^config/FellWalker_config.lis'
out='FW_results/jcmts20141026_00044_850_TILE7_outmap'
outcat='FW_results/jcmts20141026_00044_850_TILE7_outcat'
logfile='FW_results/jcmts20141026_00044_850_TILE7_FW.log'
method='FellWalker' rms=RMS VALUE
wcspar=True deconv=False shape='Polygon'
```

Let's break this down:

in = Your data file in which you want to find clumps
out = Name of the output file showing you a map of all those clumps
outcat = Name of the output catalogue containing clump properties in FITS format
logfile = Name of the output catalogue containing clump properties in TXT format
method = The algorithm we want to run
rms = The RMS noise value you measured from the map
wcspar = "True" means return the clump properties in terms of arcseconds, R.A. and dec as opposed to pixels and pixel numbers in the file.
deconv = "False" means "do not try to do any fancy deconvolution with the JCMT beam, just return the properties of the clumps directly as I see them in the map"
shape = "Polygon" means "save the physical shape of each clump as a polygon so I can overplot the clumps on my map"

List the contents of the "FW_results" directory (%ls FW_results).**Use Gaia to explore the .sdf file and look at the text file of clump properties:**

```
%gaia FW_results/jcmts20141026_00044_850_TILE7_outmap.sdf
%gedit FW_results/jcmts20141026_00044_850_TILE7_FW.log
```

Note that there are also a few ways to run FellWalker in python - but let's keep things a bit more simple for now.

Part II:

1. Now that you have downloaded the data, cropped the map, and found/categorised what FellWalker believes to be significant zones of 850 micron emission…it's time to do some real science! Let's find out which of your clumps are potentially collapsing under gravity and forming stars! For this, you'll need the following Jupyter notebook:

   https://www.eaobservatory.org/~s.mairs/ASTR351/assignments/assign2_JN.tar.gz

   Move the file to your Desktop:

```
%mv assign2_JN.tar.gz ~/Desktop
%cd ~/Desktop
%gunzip assign2_JN.tar.gz
%tar -xvf assign2_JN.tar
```

   In the lecture, we went over how we can relate 850 micron emission to the mass of a core. We also discussed the Jeans Mass. Recall that a core that is gravitationally unstable has a Mass to Jeans Mass ratio (M/M_J) greater than or equal to 1. Gravitationally stable cores have a ratio of less than 1.

   M = 0.074 x (Total Flux / 1 Jy) x (D / 100 pc)^2 x (κ / 0.01 cm^2 g^{-1})$^{-1}$ x {exp(17 K / T_d) - 1} Solar Masses

   M_J = 2.9 x (15 K / T_d) x (R / 0.07 pc) Solar Masses

   <u>Where:</u>
   Total Flux = "Sum" column of FellWalker output - but converted to units of Jy
   D = Distance in parsecs to the Orion Nebula

   κ = opacity = 0.01 cm^2 g^{-1}

   T_d = Dust temperature = 15 K
   R = Radius of the core assuming it has a perfectly spherical configuration (an assumption). This comes from the "Volume" column of FellWalker output (which is really the area of the clump since our data is 2D - so we will call it "A"). Note that if our core is spherical, its 2D projection in our map would be circular and that the area of a circle is: $A = \pi R^2$

So, we need to calculate the Mass and the Jeans Mass for each clump identified by FellWalker. In the Jupyter notebook - I show you an easy way to read all of the clump properties into Python calculate the stability of all the clumps at once. If you prefer other programming languages, or some other method using the text file version of the clump properties - please feel free to do that!

Use all of the information in the Jupyter Notebook along with the equations, above, to calculate the Jeans Stability of each clump (M/M_J). Plot a histogram of the clump stabilities in Tile 7 of the SCUBA-2 Orion Nebula Map. Find the R.A and Dec of any clumps you find to be unstable to gravitational collapse.

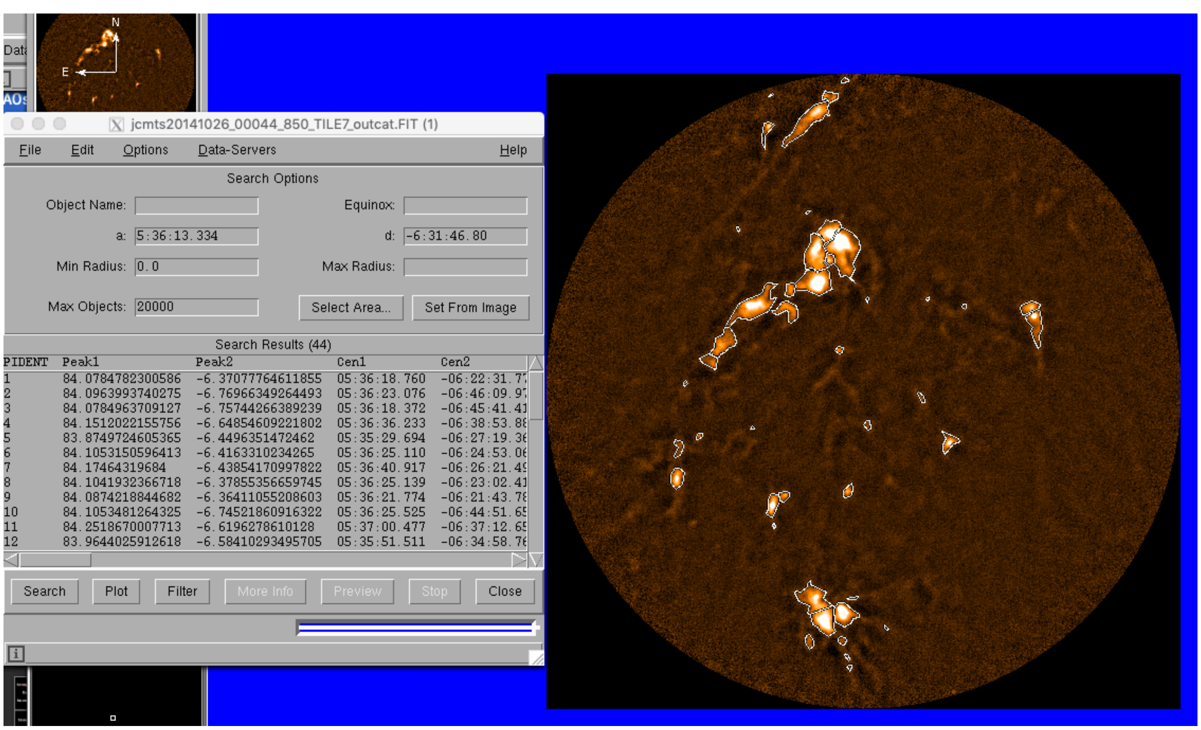
2. Now let's test the hypothesis that these "unstable" clumps are really collapsing and producing stars. To do that, we can overlay a catalogue of very young protostars (identified and characterised using Spitzer Space Telescope Infrared data) on top of our SCUBA-2 data and see if there are known, young stars in the unstable clumps.

   First, let's overlay your clumps on the data.

   ```
   %cd ~/Desktop/Clumpfinding/data/
   %gaia jcmts20141026_00044_850_TILE7_crop.sdf
   ```

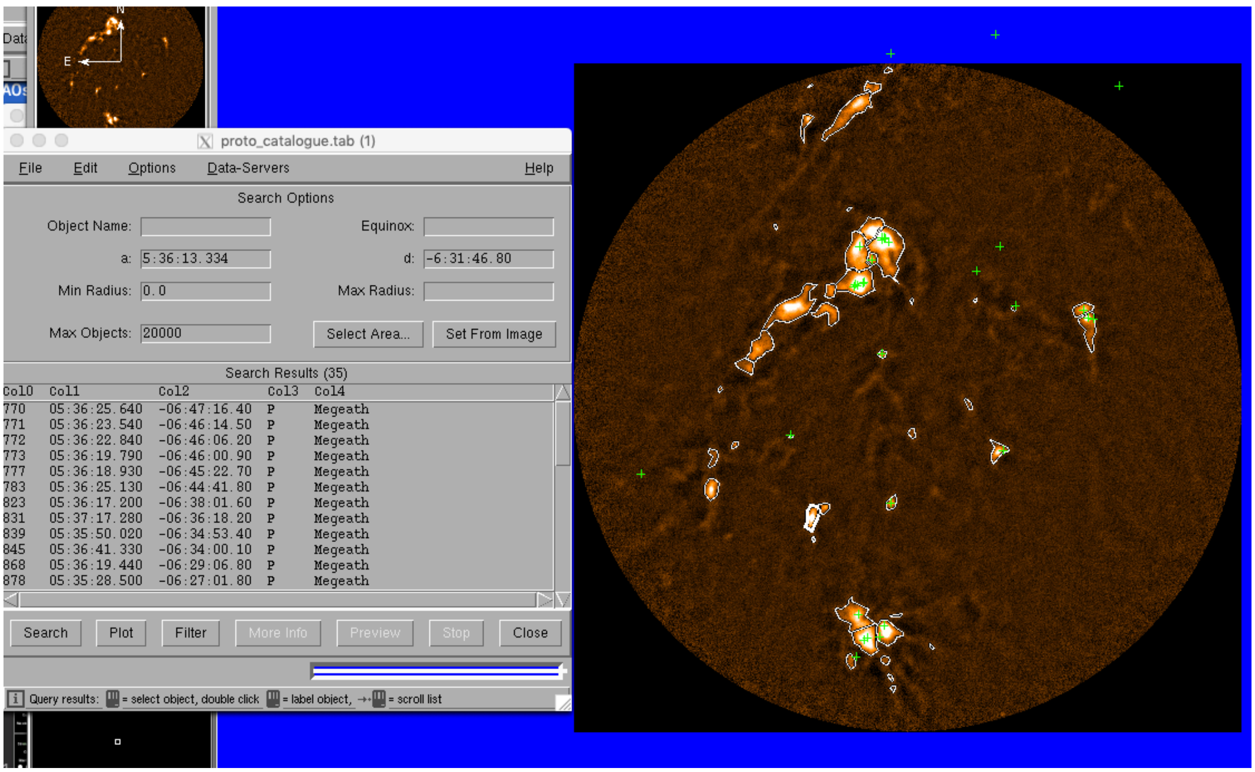
   Then, select "Data-Servers" (drop down menu), then "Local Catalogs", "Load from file", and select your clump properties files in FITS format:

   (FW_results/jcmts20141026_00044_850_TILE7_outcat.FIT)



If you click on a clump on the map, or a clump in the list that pops up - the same clump/information will be highlighted in the other window.

Then, load the YSO catalogue (~/Desktop/Clumpfinding/YSO/proto_catalogue.tab) in the exact same way as you loaded the clump catalogue. These will appear as green crosses.



Questions/Plots:

Part I

1. What are the units of the SCUBA-2 map?

2. What is the background noise in the map?

3. What radius (in arcseconds) did you crop the map down to?

4. Comment on the FellWalker.Noise parameter - is 3*RMS reasonable? Are there other values you could imagine using?

Part II

5. Attach an image of the Jeans Stability Histogram for this region with bins that clearly show the unstable and stable clumps (label the graph please!)

6. How many protostars are in the most unstable clump?

7. Attach an image of the Clumps and protostars overplotted on the SCUBA-2 image (screenshot is ok!)

8. What might explain protostars outside the boundaries of any clump?

9. What might explain an unstable (collapsing) clump without any protostars inside?

10. What might explain protostars inside clumps calculated to be Jeans stable?

References to use:

★ Slides from the lab: https://www.eaobservatory.org/~s.mairs/ASTR351/slides/ASTR351L_CF.pdf
★ Berry, D. 2015, A&C 10:22: The full FellWalker description: https://arxiv.org/pdf/1411.6267.pdf
★ FindClumps manual: http://starlink.eao.hawaii.edu/docs/sun255.htx/sun255ss5.html