



Spitzer Data Analysis Cookbook

Spitzer Heritage Archive Documentation

SSC Science User Support Team and Instrument Support Teams
IRSA Science User Support Team

Version 6.0, June 2017



Table of Contents

Purpose of this Document	5
Important Documentation.....	6
Recipe 1. Downloading Data from the Spitzer Heritage Archive	7
Recipe 2. Downloading Data from the Spitzer Heritage Archive - A Detailed Example..	23
Recipe 3. MOPEX: Mosaic IRAC S-COSMOS Data.....	36
Recipe 4. MOPEX: Mosaic IRAC Galactic FLS Data	65
Recipe 5. MOPEX: IRAC Mosaic of the galaxy NGC1097	72
Recipe 6. MOPEX: Making an IRAC Array Location Dependent Correction Mosaic...	80
Recipe 7. MOPEX/APEX: Point Source Photometry from IRAC Images	91
Recipe 8. Bandmerge: A How-To Example	109
Recipe 9. Spitzer Synthetic Photometry	122
Recipe 10. CUBISM: Spectrum Extraction of M51	133
Recipe 11. Dark_Settle: Correct Dark Currents for HR 6606	146
Recipe 12. SPICE: Command-line Extraction of HH 46/47	154
Recipe 13. IRSFRINGE: Remove fringes in data of HH 46/47	165
Recipe 14. SPICE: Spectrum Extraction of NGC4579	168
Recipe 15. SPICE: Spectrum Extraction of HH 46/47	178
Recipe 16. SPICE: Spectrum Extraction for Comet S-W 1.....	188
Recipe 17. SPICE: Spectrum Extraction of ultra-deep IRS data.....	193
Recipe 18. IRSCLEAN, COAD, and SPICE: Mrk 273 LH Data	199
Recipe 19. SMART: General Steps for Extracting Staring Mode Spectra	222
Recipe 20. SMART screen shot recipe.....	225
Recipe 21. MOPEX: Mosaic of MIPS data for HII 173	238
Recipe 22. MOPEX/APEX: MIPS 24um mosaic of a $z=0.7$ cluster	242
Recipe 23. MOPEX: PRF Estimation and Point-Source Extraction	253

Recipe 24.	MOPEX: MIPS 24 micron data from MIPS GAL	257
Recipe 25.	MOPEX: MIPS 70 micron mosaic of HII 173	265
Recipe 26.	MOPEX/APEX: MIPS 70um photometry of HII 173	268
Recipe 27.	MOPEX: MIPS 70 microns mosaic of the GFLS.....	270
Recipe 28.	MOPEX/APEX: Point-Source Photometry at 70um.....	276
Recipe 29.	GeRT: Bad Stim Flash Correction in MIPS-70 data.....	288
Recipe 30.	GeRT: Filter Extended Sources in MIPS-70 data	293
Recipe 31.	GeRT: Correct Mild Saturation in MIPS-70 data	300
Appendix A.	Version Log.....	308

Purpose of this Document

Here we present step-by-step recipes for reducing data taken by all three instruments on board the Spitzer Space Telescope. We have designed the recipes to use only publicly available data and routines so that anyone can use them at any time.

Additional information regarding the analysis of Spitzer data is available in Instrument Handbooks, the FAQ, and the Data Analysis and Tools page, all linked from the Spitzer webpage.

Important Documentation

The following documents and webpages are recommended reading, and are referred to throughout this Cookbook:

IRAC Instrument Handbook:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/iracinstrumenthandbook/>

IRS Instrument Handbook:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irs/irsinstrumenthandbook/>

MIPS Instrument Handbook:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/mips/mipsinstrumenthandbook/>

Recipe 1. Downloading Data from the Spitzer Heritage Archive

This recipe describes how to retrieve data from the Spitzer archive using the Spitzer Heritage Archive (SHA) web-based interface.

The SHA is housed at IRSA, the Infrared Science Archive. The URL is

<http://sha.ipac.caltech.edu/applications/Spitzer/SHA/>

1.1 Quick Hints, Tips, and Term Definitions:

- An individual Spitzer observation sequence is an *AOR*, or *Astronomical Observation Request*. In certain cases (often calibration or sometimes science observations), you may also see an *IER*, or *Instrument Engineering Request*. Either one involves many individual frames. Both have a large integer number called an *AORKEY* that uniquely identifies that observation within the mission.
- The individual data frames that emerge, calibrated, from the Spitzer pipeline are *Level 1*, or *Basic Calibrated Data*, or *BCDs*.
- The products that come from combining these individual data frames (such as mosaics of individual pointings) are *Level 2*, or *post-BCD*, or *PBCD data*.
- Spitzer observations can cover large areas or, by design, multiple targets. If you are interested in just portions of the larger observation, you can choose to have just *individual data products* returned -- e.g., just the observations that went into the portion of the sky for which you searched -- or you can return the products for the whole AOR.
- The SHA website displays subwindows, called *panes*. Items within each pane often appear in *tabs* – click on different tabs to display different pane contents.
- Also available are Enhanced Products that come from combining AORs or doing post processing (such as synthetic photometry from spectra or source extraction from images). These data can be a very powerful way to get started on using Spitzer data in your science pretty much straightaway. Enhanced products generated by the Spitzer Science Center (SSC) are returned in a tab separately from the substantial contributed enhanced products that were delivered by the community to the SSC and IRSA (and continue to be delivered to IRSA); these contributed enhanced products can include mosaics, photometry, spectra, and data from telescopes other than Spitzer.

1.2 Searching:

1. Go to the SHA website

(<http://sha.ipac.caltech.edu/applications/Spitzer/SHA>). The first page is the search interface. It looks like this:

2. **Query the Spitzer Heritage Archive** for the desired data set. This can be accomplished in a number of ways. By default, it comes up with a position search, as can be seen above. On the left-hand side, several additional search options appear, including a link to the position search – you can search by abstract text, moving object, observation date, AORKEY (the integer number corresponding uniquely to each AOR), program, observer, or campaign.

Searching By Position: You can enter the name of the object and have the machine try either NED or SIMBAD first to resolve the coordinates, or you can enter the coordinates directly.

For this example, use:

RA 03 43 48.38 (J2000)

Dec +25 00 15.23 (J2000)

Which you can type in directly as : 03 43 48.38 +25 00 15.23

The SHA echoes back to you what it thinks you have told it, right below the box in which you type the name or coordinates. If you enter a name, by default it tries "NED first then SIMBAD" – if it incorrectly resolves it, select "SIMBAD first then NED" and it should re-resolve the coordinates in real time. If that is still wrong, enter the coordinates directly.

The search radius also can be changed, or you can leave it as the default. For purposes of this example, change the "Radius (arcsec)" field to 5 arcmin by selecting "Arc Minutes" from the pulldown and entering 5 in the box.

The next lines in the search window ask which kinds of data you want displayed: *Display search results in tabs for: Observation Request (AOR), Level 2 (PBCD), Level 1 (BCD)*, and for the position search, also both *SSC Enhanced Products (IRS), SEIP Super Mosaics, SEIP Source List and Contributed Enhanced Products*. Several of the other searches will allow you to select “SSC Enhanced Products” but not “Contributed Enhanced Products.” By default, all searches return a list of AORs under the "AOR" tab, and a list of Level 2 products under the "Level 2 (PBCD)" tab (see figure below). You can also choose to access the original BCD frames by selecting the "Level 1 (BCD)" option here, and then the corresponding tab will be returned by your search. Similarly, if you choose to search for enhanced products, those tabs may be returned; any given search may not have any enhanced products to return – while there are IRS enhanced products scattered all over the sky, and while we have had the community return back to us products from observations all over the sky, there is much of the sky not covered by these observations. For this example, ensure that at the very least, the AOR and Level 2 (PBCD) options are checked, but you may also check both of the Enhanced Product boxes (SSC Enhanced Products (IRS), and Contributed Enhanced Products).

Finally, there is a “More options” line. If you click on this, additional options will appear. You can choose to apply the spatial constraints to the AORs or individual data products – what this means is the following. AORs can cover large areas or, by design, multiple targets. You can choose to return all AORs touching your search area, or just the individual data products enclosed within your search area – e.g., just the observations that went into the portion of the sky for which you searched. For this example, check “AORs”. The next portion of the options allows you to search by instrument or wavelength, and, at the bottom, individual frame time and time of observation. You may be interested in just the MIPS data, in which case you should unclick the “All IRAC” and “All IRS” boxes, and leave the “All MIPS” box checked. For this example, leave it all as the defaults.

See the figure below for what your screen should look like at this point.

Position Search

General

- [Position](#)
- [Abstract Text](#)

Advanced

- [AORKEY](#)
- [Campaign](#)
- [IRS Enhanced Products](#)
- [Moving Object](#)
- [Precovery](#)
- [Observation Date](#)
- [Observer](#)
- [Program](#)

Position From File

Name or Position: Try NED then Sin

55.951583, 25.004231 Equ J2000 or 3h43m48.38s, +25d00m15.2s Equ J2000

Radius: Arc Minutes

Display search results in tabs for:

Observation Request (AOR)
 SSC Enhanced Products (IRS)
 Contributed Enhanced Products
 Level 2 (PBCD)
 SEIP Super Mosaics
 Level 1 (BCD)
 SEIP Source List

More options

Apply spatial constraints to: AORs Individual data products

Select the filter to search by: Instrument Parameters Wavelength Range

All IRAC

IRAC Map

IRAC Map PC

3.6 μm

4.5 μm

5.8 μm

8.0 μm

Readout Mode

Full Array

Stellar Mode

High Dynamic Range

Subarray

All IRS

IRS Stare

IRS Map

IRS Pickup Image

Hi Res 9.9-19.6 μm

Hi Res 18.7-37.2 μm

Low Res 5.2-8.7 μm

Low Res 7.4-14.5 μm

Low Res 14.0-21.7 μm

Low Res 19.5-38.0 μm

PU Blue 13.3-18.7 μm

PU Red 18.5-26.0 μm

All MIPS

MIPS Phot

MIPS Scan

MIPS SED

MIPS TP

24 μm

70 μm

160 μm

Photometry Scale

Fine (70 μm only)

Default (70 μm only)

Scan Rate

Fast

Medium

Slow

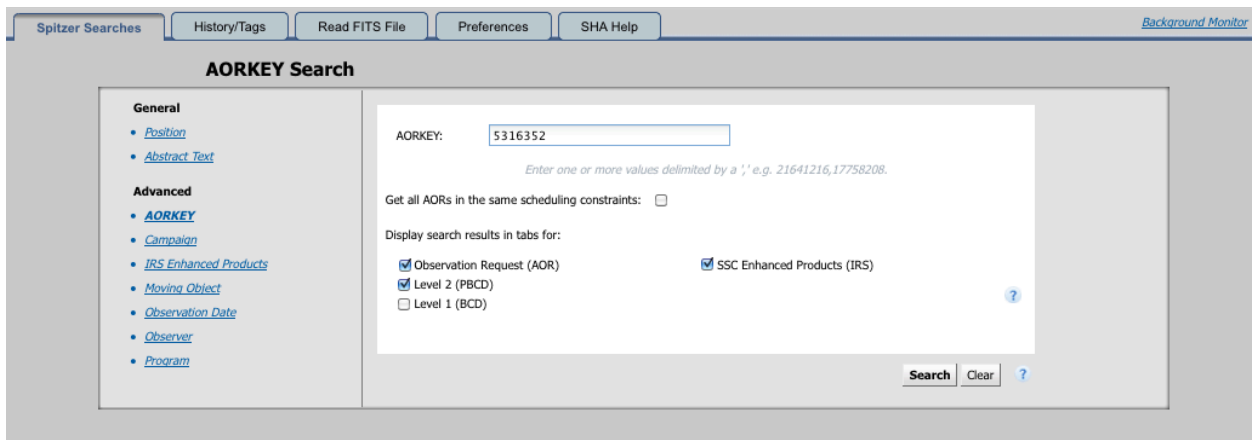
Frame Time Min (seconds): Frame Time Max (seconds):

Start Date (UT): End Date (UT):

Search
Clear
?

Then click on "Search" at the bottom of the browser window. The SHA will return 26 AORs for your consideration. You can now proceed to Step 3 below.

By AOR ID: Selecting "search by AORKEY" from the left array of options pulls up a very short window asking for the AORKEY. For this example, type in the ID number 5316352. As for the other search options, you have choices for which tabs are returned by the search. Note that 'Contributed Enhanced Products' is not an option in this search because the contributed enhanced products are not necessarily tied to AORKEYs. For this example, ensure that at least the AOR and Level 2 options are checked; you can also select SSC Enhanced Products (IRS). Click search.

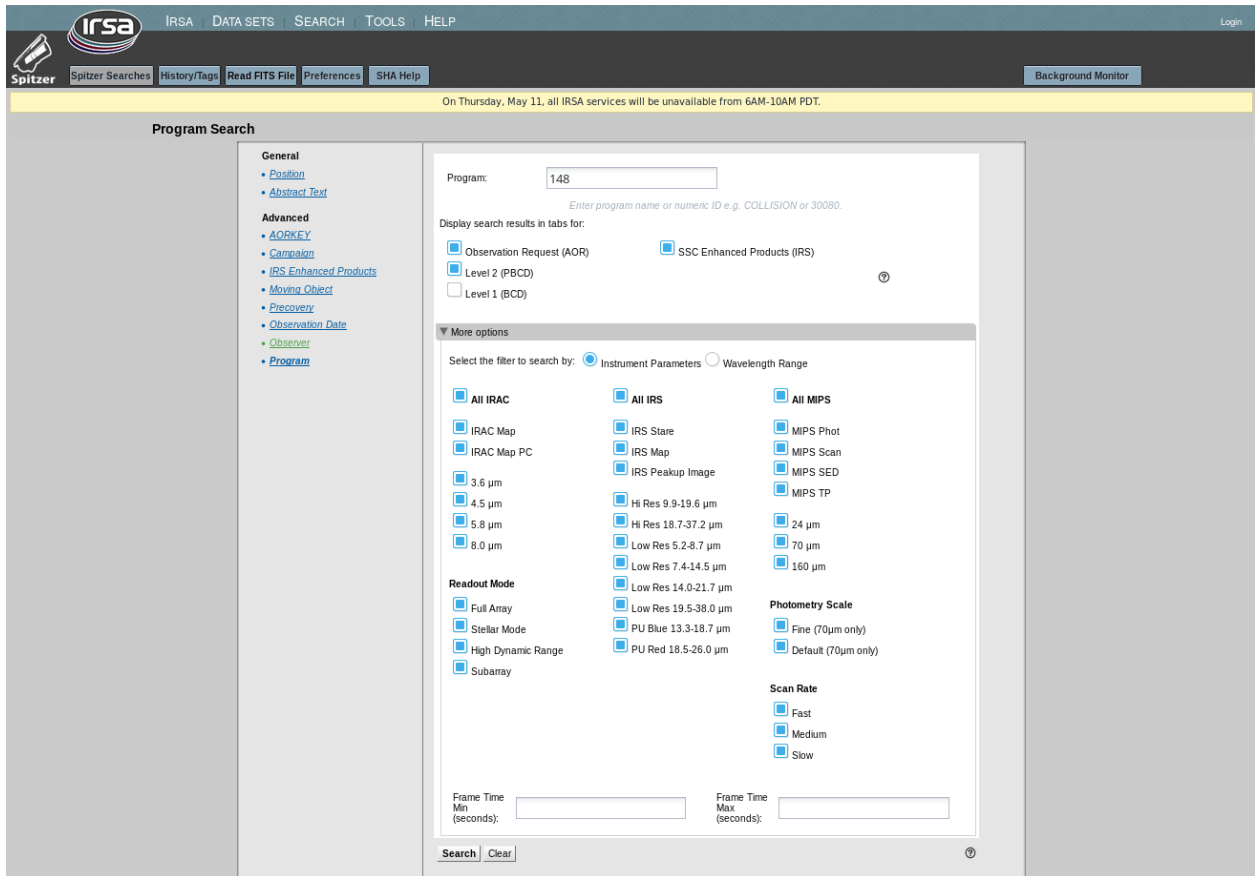


Only one AOR will appear in the SHA search results pane. You can now proceed to Step 4.

By Program: Selecting “search by program” from the left array of options pulls up a shorter form, but with common elements to the position search. For this example, enter program "148"; you could also enter the program’s name. As above, you have choices for which tabs are returned by the search. Note that ‘Contributed Enhanced Products’ is not an option in this search because the contributed enhanced products are not necessarily tied to programs. For this example, ensure that at least the AOR and Level 2 options are checked; you may also check SSC Enhanced Products (IRS).

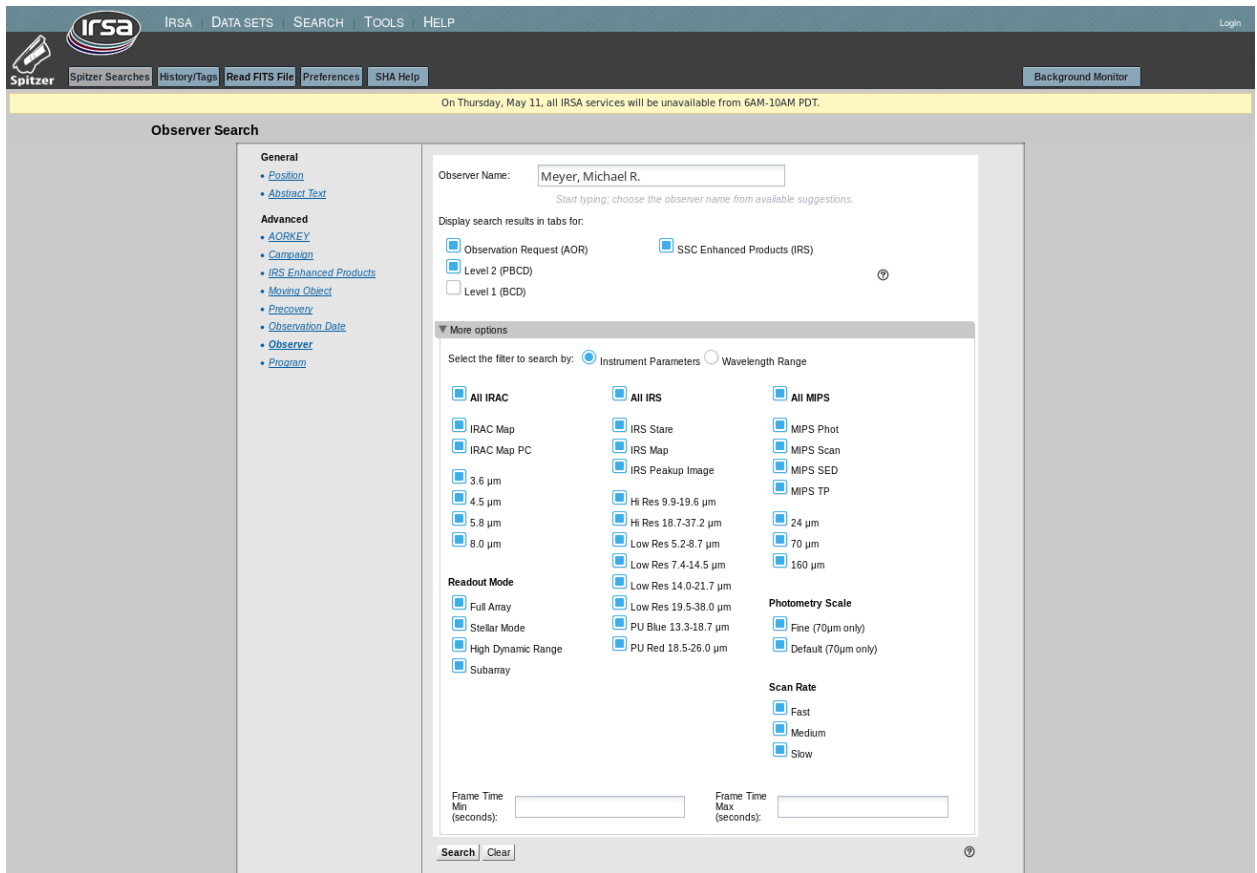
As above, if you click on “More options”, you can choose to only display a subset of the data. Leave all of them on for purposes of this example.

Then click on "Search" at the bottom right-hand of the window.



The AORs associated with this program will appear in the SHA search results pane. You can now proceed to Step 3.

By Campaign/Observer: Selecting “search by Observer” from the left array of options pulls up a similarly short window. If you type in "Meyer", it will give you a list of options to auto-complete your selection. Pick Michael Meyer’s name. As before, you have choices for which tabs are returned by the search. Note that ‘Contributed Enhanced Products’ is not an option in this search because the contributed enhanced products are not necessarily tied to an observer. For this example, ensure that at least the AOR and Level 2 options are checked; you may also check SSC Enhanced Products (IRS). As above, if you click on “More options”, you can choose to only display MIPS data. For this example, leave them all checked. Click on “Search” in the lower right.



Many AORs will appear in the SHA search results pane. You can now proceed to Step 3.

3. **Find the AOR you want in the list** of returned AORs under the AOR tab. For this example, the one you want is observing a target called “HII_174” and is located at position (decimal RA, Dec) 55.951375 25.004417, or 03h43m48.33s +25d00m15.90s.

Search results from the position search above look like this:

Target name	RA (J2000)	Dec (J2000)	NAIF ID	Instrument/Mode
Pleiades	3h47m00.00s	+24d07m00.0s		MIPS Scan
Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map PC
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map PC
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map PC
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map PC
Pleiades-MS	3h44m37.32s	+25d08m17.3s		IRAC Map PC
Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map PC
Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map PC
HII_174	3h43m48.33s	+25d00m15.9s		IRS Stare
HII_174	3h43m48.33s	+25d00m15.9s		MIPS Phot
HII_250	3h44m04.24s	+24d59m23.4s		IRS Stare
HII_250	3h44m04.24s	+24d59m23.4s		MIPS Phot
RX_J0344.3+2447	3h44m20.08s	+24d47m46.2s		MIPS Phot
HII_250	3h44m04.24s	+24d59m23.4s		IRAC Map
HII_174	3h43m48.33s	+25d00m15.9s		IRAC Map
HII_173	3h43m48.41s	+25d11m24.2s		IRAC Map
RX_J0344.3+2447	3h44m20.08s	+24d47m46.3s		MIPS Phot
Pleiades	3h44m36.00s	+25d24m00.0s		MIPS Scan
Pleiades	3h44m36.00s	+25d24m00.0s		IRAC Map
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map
Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map
Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map
Pleiades-MS	3h44m37.32s	+25d08m17.3s		IRAC Map
Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map
Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map

Name	Value
AORKEY	8798720
Scan rate	medium
Require 160	1
Fast reset 160	
Step size forward(")	160
Step size turnaround(")	160
No. of scan legs	12
Map cycles	1
Scan leg length(")	1.000000
Map cross-scan offset(")	900
Map in-scan offset(")	0
Backward map	0
QA comments	STATUS = " Nominal/OK to Release"; PROCES
Campaign name	MIPS002800

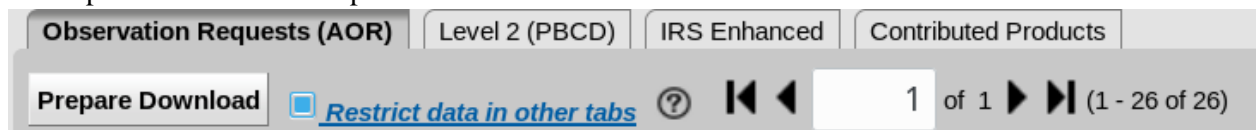
Things to note about this search result:

- The type of search you are using is summarized at the top – in this case, a position search. To perform a different search, click on the “Spitzer searches” blue tab at the top left.
- Your search results appear in two panes – search results and details. On the left are the search results in one pane, with an AOR tab and a Level 2 (PBCD) tab, and in this case “IRS Enhanced” and “Contributed Products” tabs; on the right is the “Details” pane, with tabs: Details, AOR DoC, and AOR Footprint. The contents of the Details pane changes corresponding to what you have selected (e.g., which tab and which green highlighted line) in the search results pane; more on this below.
- If there are observations found that are still proprietary, e.g., not yet public, they will be highlighted by a red background color in the search listing. You cannot examine or download those data, unless you are (a) logged in, and (b) authorized by the PI to have access to those data; contact the Helpdesk if you are in this situation and still do not have access to those data. Data become public at most a year after they were obtained, so in the worst case, you just have to wait.

- These search results were found via the “search by position” example above, so if you did a different search, you may have more or fewer tabs and more or fewer or different AORs listed; more on this next.

Recall that the object we are looking for is a target called “HII_174” and is located at 3h43m48.33s, +25d00m15.9s. Depending on exactly how you got to the results pane, you may need to find this object among many AORs. In order to do this, you have several options. Note first how many search results you have to sift through; at the top of the search results pane, it tells you how many AORs are returned. The default number of AORs displayed per page is 50, and you can adjust this as your needs dictate. It is not generally recommended to have 1000s of AORs shown per page, because it will take too long for your browser to render the table.

The top of the search results pane looks like this in the search results above:



In this case, there are only 26 AORs shown (“1-26 of 26”). If you searched by PI name without restriction on the instrument, you could have more than 1000 AORs in the list “(1-50 of 1006)”.

To find your desired AOR among many, you can do any of several things, including:

- Click on a column heading to sort by that column heading, then look for your AOR in the list. For example, in this case, we know our desired target is called “HII_174”, so we could sort by Target name and then find the page on which this target appears.
- Impose a filter. Click on the “Filter Panel” (the funnel-looking symbol) at the top right of the results pane. You may need to grab the slider between panes and slide it over to see this. The slider is the divider with blue left-right arrows between the panes, annotated with a black arrow in this figure:

Position 55.95158333333334;25.004230555555555;EQ_J2000 with a radius of 0.0013 deg

Observation Requests (AOR) | Level 2 (PBCD) | IRS Enhanced

Prepare Download Restrict data in other tabs

Target name	RA (J2000)	Dec (J2000)	NAIF ID	Instrument/Mode	AORKEY	AOR label	Observation start time	Observation end time
<input checked="" type="checkbox"/> HII_174	3h43m48.33s	+25d00m15.9s		IRS Stare	5316096	IRSS-S180	2004-09-26 10:48:03	2004-09-26
<input type="checkbox"/> HII_174	3h43m48.33s	+25d00m15.9s		MIPS Phot	5316352	MIPS-S180	2004-09-22 08:35:49	2004-09-22
<input type="checkbox"/> HII_174	3h43m48.33s	+25d00m15.9s		IRAC Map	5315840	IRAC-S180	2004-09-02 13:12:15	2004-09-02
<input type="checkbox"/> HII_250	3h44m04.24s	+24d59m23.4s		MIPS Phot	5317120	MIPS-S181	2004-09-22 08:54:36	2004-09-22
<input type="checkbox"/> Pleiades	3h44m36.00s	+25d24m00.0s		MIPS Scan	6526720	Pleiades_2_map	2004-09-23 21:56:01	2004-09-24
<input type="checkbox"/> Pleiades	3h44m36.00s	+25d24m00.0s		IRAC Map	6526464	Pleiades_2_IRAC_map	2004-09-10 03:12:24	2004-09-10
<input type="checkbox"/> Pleiades	3h47m00.00s	+24d07m00.0s		MIPS Scan	8798720	Pleiades_1_mapB	2004-02-23 23:17:46	2004-02-24
<input type="checkbox"/> Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map PC	42494208	IRAC-0004-short MP	2011-11-14 19:29:10	2011-11-14
<input type="checkbox"/> Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map PC	42492928	IRAC-0003-long-set1 MP	2011-10-22 14:32:14	2011-10-22
<input type="checkbox"/> Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map PC	42493440	IRAC-0003-short MP	2011-10-22 14:09:00	2011-10-22
<input type="checkbox"/> Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map PC	42493184	IRAC-0003-long-set2	2012-03-27 15:40:12	2012-03-27
<input type="checkbox"/> Pleiades-Bds	3h41m33.70s	+23d11m44.7s		IRAC Map	3970816	IRAC-0004-short MP	2005-02-18 22:30:46	2005-02-18
<input type="checkbox"/> Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map	3970048	IRAC-0003-short MP	2004-02-14 21:01:28	2004-02-14
<input type="checkbox"/> Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map	3969792	IRAC-0003-long-set2	2006-02-07 01:40:55	2006-02-07
<input type="checkbox"/> Pleiades-Bds	3h44m08.90s	+24d38m07.2s		IRAC Map	3969536	IRAC-0003-long-set1 MP	2004-09-09 04:08:17	2004-09-09
<input type="checkbox"/> Pleiades-MS	3h44m37.32s	+25d08m17.3s		IRAC Map PC	42495488	IRAC-0010-ms-12s MP	2004-09-12 17:00:53	2004-09-12
<input type="checkbox"/> Pleiades-MS	3h44m37.32s	+25d08m17.3s		IRAC Map	3972096	IRAC-0010-ms-12s MP	2004-09-12 17:00:53	2004-09-12

Additional Information

Name

AORKEY

Peakup mode

Peakup accuracy

Peakup filter

Peakup flux density

Extended source

Peakup position

Peakup RA offset

Peakup Dec offset

Peakup RA

Peakup Dec

Peakup RA PM

Peakup Dec PM

Click on the “Add filters” box at the top right of the results pane. Spaces will appear below column headers and you can enter an operator (if needed) and a value to compare. For this example, enter in the AORKEY field “5316352” and type “enter” or “return”. (You could impose more than one filter in this process.) For this case, only one AOR should remain in the list after you impose the filter. Filters are a powerful way of weeding down the list, and there are more examples of using them in the SHA manual.

4. **Get ready to download the data and do it.** Click on the checkbox on the far left of the row associated with this AOR (AORKEY 5316352, target “HII_174”). Click on the “Prepare Download” button above this column in the search results window. A popup appears asking what kind of data you want. For this example, select the “Level 2, including ancillary” and “Level 1, including ancillary” products. No need to download the calibration data or raw data.

Observer Meyer, Michael R.

Observation Requests (AOR) | Level 2 (PBCD) | IRS Enhanced

Prepare Download Restrict data in other tabs

Target name	AORKEY
<input checked="" type="checkbox"/> HII_174	5316352

Download Options

Choose any related data to include with the observation request data.

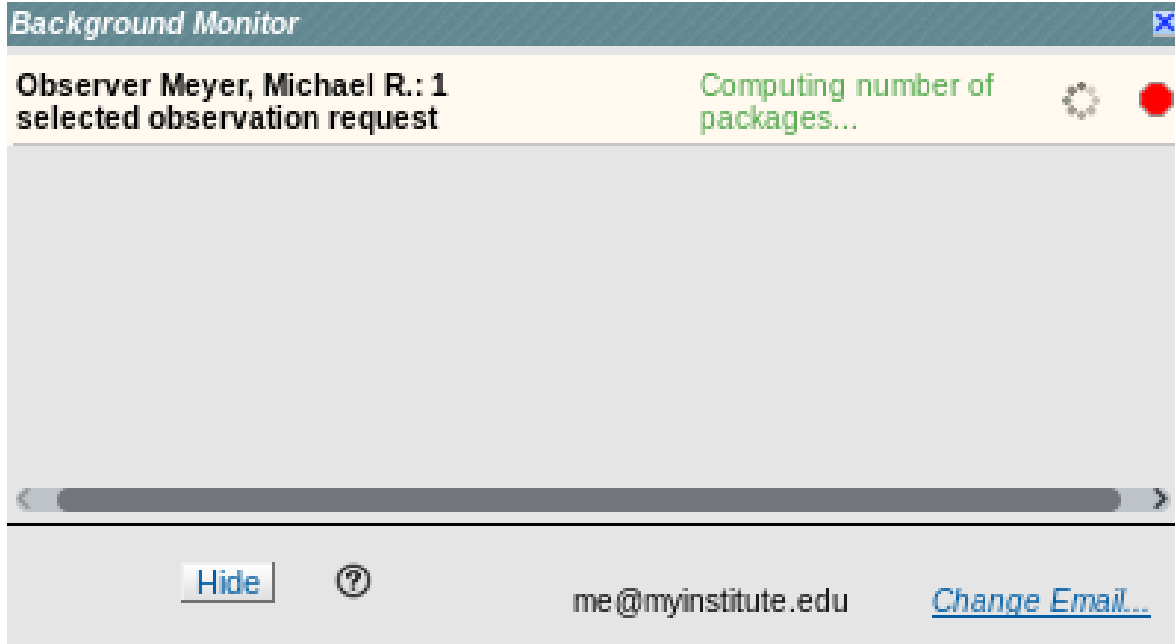
- Level 2 products (PBCD)
- Including ancillary
- Level 1 products (BCD)
- Including ancillary
- Calibration data
- Raw data products

Send me email with URLs to download

Email:

Prepare Download Cancel

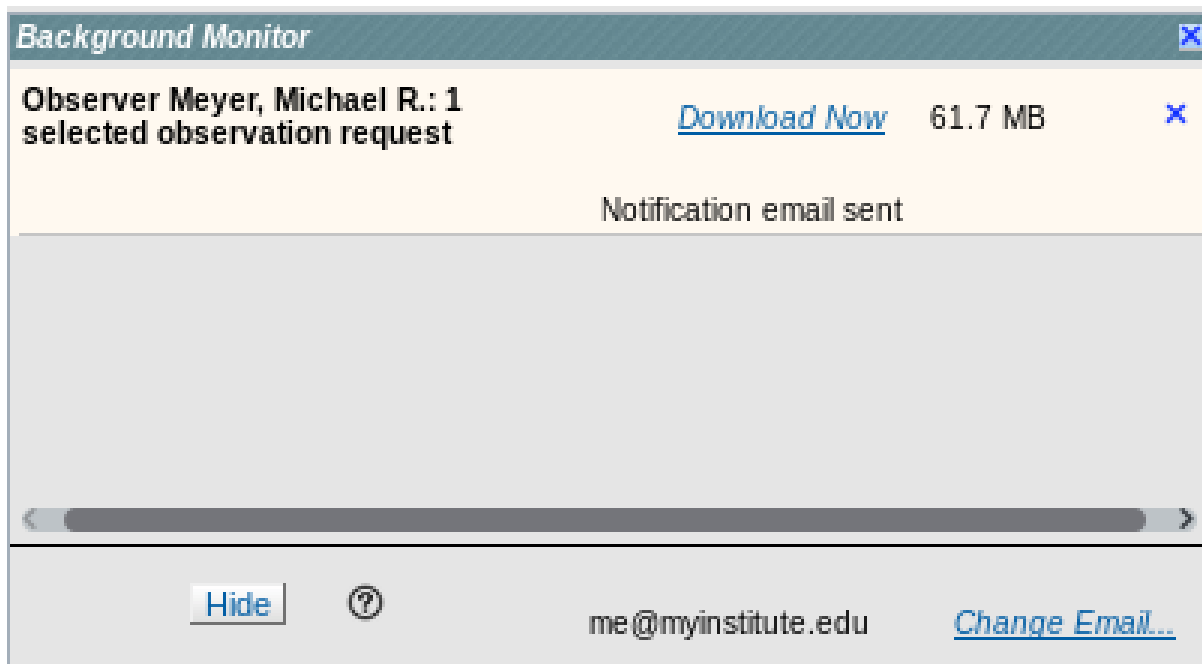
The total compressed data set takes up ~62 MB and should package very quickly. On some browsers, it may be so small that it downloads all by itself. The “Background monitor” will appear as a link on the top right of the SHA window, and you can watch it package if you click on it and look quickly.



When it has completed, if the Background Monitor is not open, it will tell you it is ready:



When you open the Background Monitor, a “Download Now” link appears; click on it to download it.



Your browser may put this file automatically somewhere on your disk, or it may ask you where you want it saved. This is a local, configurable option, controlled in your browser preferences. If you can not find the download, look for recently modified files on your disk or look for a folder called something to the effect of “Downloads.”

If you have a very big set of data to package up, and you did not enter an email when you did the packaging, you can ask during the packaging process for an email to be sent to you when it is done. Click on “Add Email...” in the lower right of this pop-up window in order to add an email; click on “Change Email” if you notice midstream that you have a typo or would like to change the address.

5. **Unzip the files that you have saved to your disk if your browser did not do it for you.** Your filename will vary, depending on exactly what search you did to get here. You can unzip by double-clicking on the file, or type this in a terminal window:

```
unzip req-ID-5315584-1-selected_AORs.zip
```

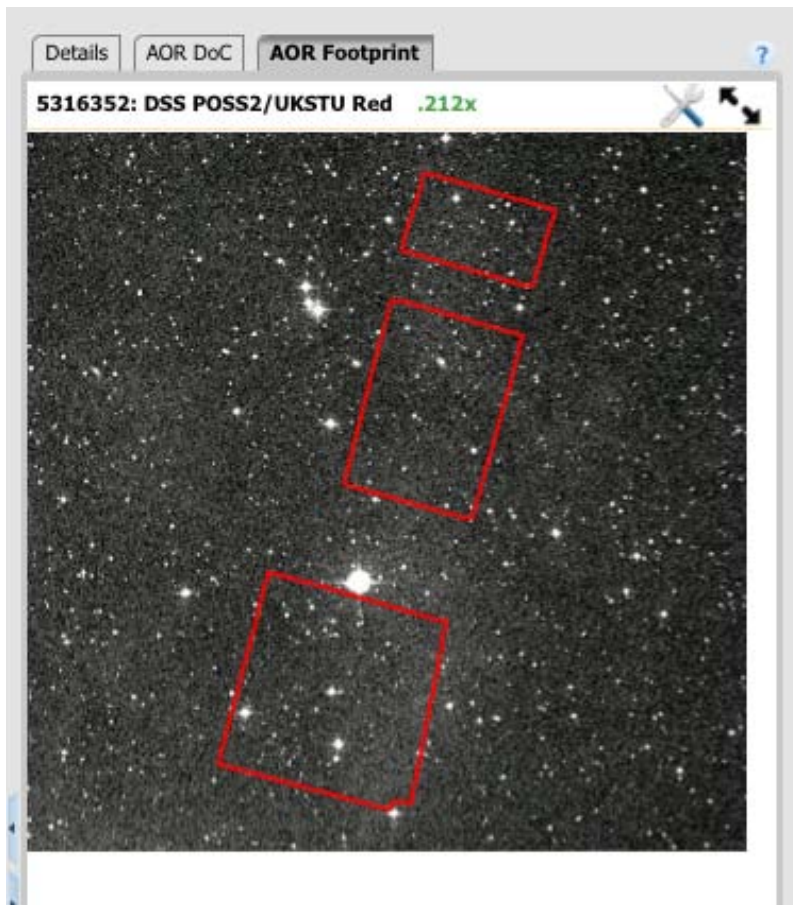
6. **Figure out what all of these files are.** Check the Spitzer web site for detailed documentation on data filename conventions and of course the instrument handbooks.

The data we have just downloaded may be easier to understand if we investigate the details of the observation itself.

In the main SHA search results window, select this observation. (It may very well be the only one left in your search results window at this point.) On the far right of the SHA window, there is a “Details” pane which contains many details about the observation. The first tab, “Details”, gives you the original

abstract and observer, as well as the instrument parameters that went into this specific observation, such as frame time. The second tab, “AOR DoC”, is the AOR depth of coverage, e.g., how many individual frames were obtained per patch of sky for this observation. (Note it is number of frames, not total number of seconds.) The third tab is “AOR footprint”. Select that.

By default, the SHA attempts to pick an “appropriate” background image. For this observation, it selects a IRAS or DSS background image, and overlays red polygons on top of it. The red polygons outline the borders of the complete AOR. The annotation at the top right of the image tells you the AORKEY that is overlaid, the nature of the background image (DSS POSS2 in this case), and the zoom factor (0.212x in this case).



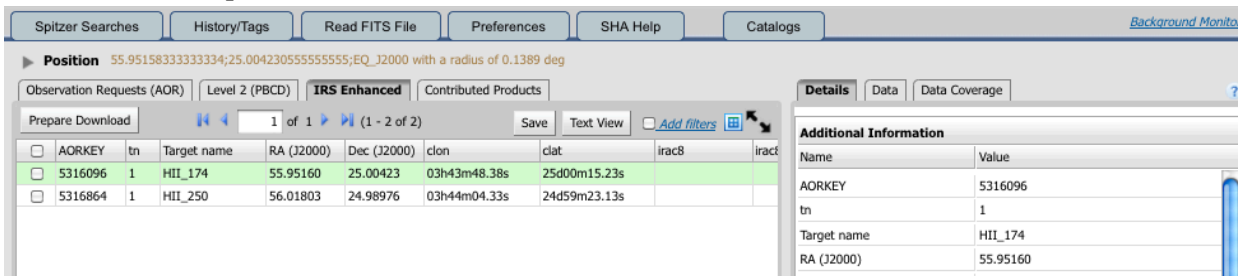
If you let your mouse hover over the image, you will get a popup window with information about the image pixel underlying your mouse. You will also see one icon appear on the top right – the one with arrows will “pop out” the image such that it takes up your whole browser window; there are also several tool icons along the top that allow you to manipulate the image (e.g., change the color table). See the online SHA help for more information on each of the options, or let your mouse hover over an icon to get a “tool tip.” Depending on what other visualizations you have been doing, the “pop out” of your image may show you your current as well as previous images. Select from the multi-view icons at the top of the popped out window to show more or fewer images; click on the “Close” arrow at the upper left to return to the search results view.



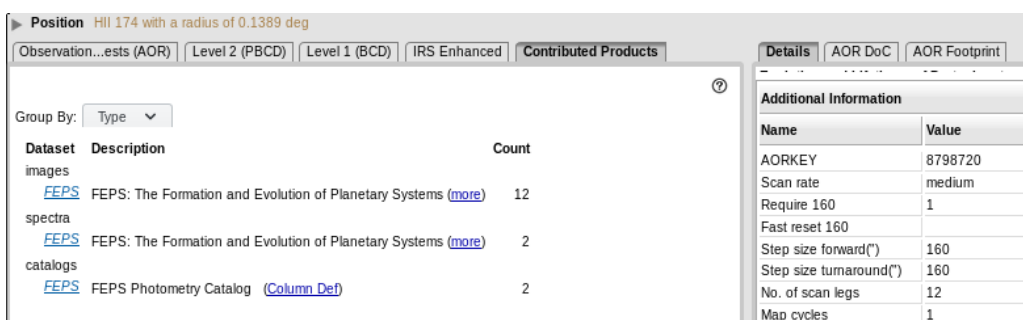
7. **Enhanced Products.** If you selected either of the Enhanced Products options, you should have additional tabs to examine. By design for this example, this observation was part of FEPS; FEPS, as a Spitzer Legacy project, was a large, cohesive observing program and also delivered observations back to the SSC. Also by design, this region also contains IRS Enhanced Products. If you selected both of those checkboxes for a position search, you should have those tabs as options:



Not all regions on the sky will return IRS Enhanced or Contributed Products. If your search does not return any data, no data will be shown in those tabs. For this example, though, you should find this for IRS Enhanced products:



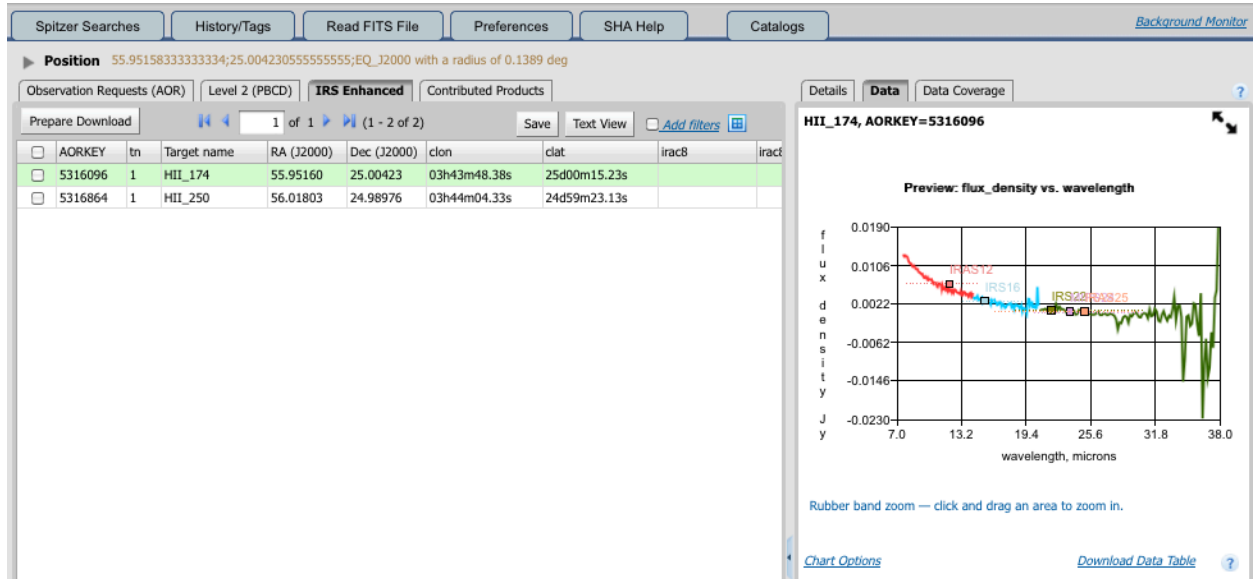
For the IRS Enhanced Products, there is more information about those products in the IRS Instrument Handbook. You should find at least this for the Contributed Products:



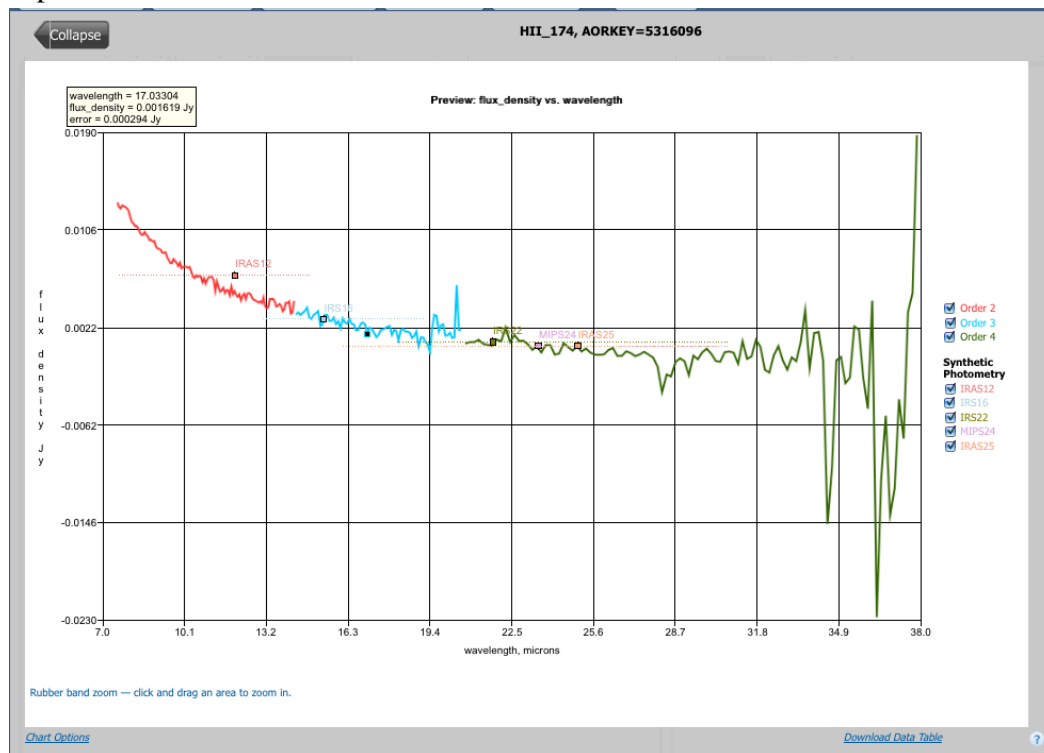
So, FEPS delivered 12 images, and 2 spectra. They also delivered some photometry in two catalogs. To learn more about the deliveries (e.g., what exactly is in the delivery, quality flags, etc.), click on "more" (link on the right). The "Data Delivery Document" is the main documentation accompanying the delivery. To see what was delivered, click on the team name (link on the left).

In both cases, the "Details" pane on the right allows you to examine the data in more detail.

For the IRS Enhanced products, you can interactively explore the spectrum.



If you click on the two black arrows in the upper right of the spectrum viewer, you can get an expanded view:



A pop-up shows the values of the spectrum under your mouse.

For the Contributed Enhanced Products search results, this is a summary of all of the results it found. To see more, first click on a program's name. It will spawn a tab that lists all of the contents, images in this case, that it found consistent with your search results:

The screenshot shows the Spitzer Data Analysis software interface. At the top, there are navigation buttons: Spitzer Searches, History/Tags, Read FITS File, Preferences, SHA Help, and Catalogs. A status bar indicates the current position: 55.9515833333334;25.0042305555555;EQ_J2000 with a radius of 0.1389 deg. Below this, there are tabs for Observation Requests (AOR), Level 2 (PBCD), IRS Enhanced, Contributed Products, and FEPS_images. The FEPS_images tab is active, showing a table of search results. The table has columns for RA (J2000 deg), Dec (J2000 deg), RA (J2000), Dec (J2000), and instrume. The first row is highlighted in green. To the right of the table is a 'Data Coverage' pane showing a star field with red and blue rectangular outlines representing data products. The selected row from the table is highlighted in blue in the data coverage pane.

RA (J2000 deg)	Dec (J2000 deg)	RA (J2000)	Dec (J2000)	instrume
55.9505805	25.1890004	03h 43m 48.14s	+25d 11m 20.4s	MIPS
56.0165601	24.9888297	03h 44m 03.97s	+24d 59m 19.8s	MIPS
55.9523437	25.1898232	03h 43m 48.56s	+25d 11m 23.4s	MIPS
56.0175161	24.9902112	03h 44m 04.20s	+24d 59m 24.8s	IRAC
55.9514350	25.0047342	03h 43m 48.34s	+25d 00m 17.0s	IRAC
55.9502710	25.0034226	03h 43m 48.07s	+25d 00m 12.3s	MIPS
56.0182846	24.9895833	03h 44m 04.39s	+24d 59m 22.5s	MIPS
55.9519797	25.0041696	03h 43m 48.48s	+25d 00m 15.0s	MIPS
56.0173495	24.9902917	03h 44m 04.16s	+24d 59m 25.1s	IRAC
56.0173088	24.9902547	03h 44m 04.15s	+24d 59m 24.9s	IRAC
55.9511466	25.0047580	03h 43m 48.28s	+25d 00m 17.1s	IRAC
55.9511767	25.0047198	03h 43m 48.28s	+25d 00m 17.0s	IRAC

If you select the “Data Coverage” out of the Details pane on the right, it will show an outline of all of the data products shown in the results pane, with the row you have selected (green highlights on left) also highlighted in the data coverage pane (blue highlights on right).

For both of these enhanced products, you can visualize the data and select data for download in much the same fashion as for the rest of the SHA. The table columns, especially for catalogs, can be cryptic, and are as provided by the teams. Consult the Data Delivery Document (or the IRS Instrument Handbook) for more information.

Recipe 2. Downloading Data from the Spitzer Heritage Archive - A Detailed Example

This section describes how to retrieve data from the Spitzer archive using the Spitzer Heritage Archive (SHA) web-based interface. In comparison to Recipe 1, it covers more about using the SHA tools to assess the observation to see if it meets your needs.

In this recipe, you will learn how to:

- Download IRAC imaging data for CG4 using the SHA.
- Use the SHA to search the Spitzer Archive for all possible and relevant CG4 observations.
- Use the SHA to assess which observation will most quickly yield an image.
- Select data for download, and do it.

2.1 Introduction and Terminology

The Spitzer Heritage Archive (SHA) is the permanent home for all of the data collected during the Spitzer mission, plus all the documentation you need to understand these data.

The SHA is a web-based interface to the Spitzer archive, and can be accessed here:

<http://sha.ipac.caltech.edu/applications/Spitzer/SHA/>

In this section, we will go over some terminology that you will need to know to use the SHA. For lots more information, please see the SHA online help, which can be accessed from the menu bar on top (SHA Help).

An individual Spitzer observation sequence is an Astronomical Observation Request (AOR). In certain cases (often calibration or sometimes science observations), you may also see an Instrument Engineering Request (IER). Either one involves many individual frames, as well as observer name, date of observation, object or area of the sky observed, and instrument used (IRAC, MIPS, or IRS)-- these are all part of the AOR. All of Spitzer's operations (planning, scheduling, processing) have been centered around these units (AORs or IERs).

Raw data that are fundamentally unprocessed are "Level 0" data. In general, most users need never worry about Level 0 data.

The individual data frames that emerge, calibrated, from the Spitzer pipeline are "Level 1," or "Basic Calibrated Data," or "BCDs." You can get just the BCDs from a region that you want, or you download the whole AOR.

The products that come from combining these individual data frames (such as mosaics) are "Level 2," or "post-BCD," or "PBCD data." These still exist fundamentally on an AOR level, e.g., you cannot get a Level 2 mosaic that is just a portion of an AOR.

You can also get higher level data through this interface. Such products are supplemental data that are produced either by the SSC or donated to us by professional astronomers, and represent additional processing. For example, a mosaic might combine data from 7 AORs into one big mosaic, with customized (as opposed to hands-off pipeline) processing of image artifacts. If you are familiar with the Legacy Enhanced Products, these are examples of such data. It used to be that you had to go through other interfaces to get to these products. Now, you can get to them via the "Enhanced Product" Search options, available as check-boxes for many of the searches. This search option will provide any additional data currently available via the IRSA/Gator tool for the region you have selected. For this example, however, there are no enhanced products of any sort available, so it does not matter if you select them or not. For an example of a search that does return results from these resources, please see the previous chapter.

All of the images come in single-plane FITS format. The other format for some data is IPAC table files (.tbl extension). IPAC table format is really just plain text, with a special header.

Step 1. Go to the SHA.

Go here: <http://sha.ipac.caltech.edu/applications/Spitzer/SHA/>

This first page is the search page for the SHA. It defaults to a Position Search. Help is available under the "SHA Help" button.

Step 2. Position search by name

You are dropped by default into a "position search." (There are other search options on the left.) Because the object we care about (CG4) is a Galactic, not an extra-galactic, object, SIMBAD (rather than NED) is a better name resolver. By default, it comes up with "Try NED first then SIMBAD" as the coordinate resolution option. If you just type in "CG4", and don't change anything else, it will go to NED first, and find a *galaxy* by the name of CG4, at +31 degrees declination – you can see this under the "name or position" box, where the SHA resolves the name in real time. This is NOT what we want. Either before you type in "CG4" or even after you have done so, pick "Try SIMBAD first then NED" from the pulldown. It will then go and get the coordinates in real time, at the correct location, near –46 degrees declination. (Wanting to search for a whole bunch of targets? See the online help for more information on the file format.)

The default search radius is 500 arcseconds. That is fine for this example.

We want the AOR tab and the Level 2 tab to be returned, so we check off both of those, and leave "Level 1" unchecked. It does not matter if you check either of the Enhanced Products boxes – there are no data available in either enhanced product archive.

We would like to further restrict our search by instrument, so click on "More options."

We would like to apply the spatial constraints to the whole AOR, so we ensure that that radio button is checked, rather than "Individual data products."

We only want imaging data, so we can uncheck "All IRS", leaving just IRAC and MIPS. For the purposes of this example, we will also uncheck "MIPS", leaving just IRAC. If we happen to forget which wavelengths IRAC, IRS, and MIPS cover, we can also search by wavelength. And, we can also search by date of observation and frame time, if we want. For this example, it is fine to leave these blank.

THEN click "Search". This is what the relevant portion of the screen looks like right before clicking "Search":

The screenshot displays the "Position Search" interface. At the top, there are navigation tabs: "Spitzer Searches", "History/Tags", "Read FITS File", "Preferences", and "SHA Help". The main content area is titled "Position Search" and contains a search form. The "Position" tab is selected, and the "Name or Position" field contains "cg4". A "Try Simbad then NED" button is next to the field. Below the field, the results show "cg4 resolved by Try Simbad then NED" with coordinates "113.538, -46.905 Equ J2000 or 7h34m09.12s, -46d54m18.0s Equ J2000". The "Radius" is set to "500" with a unit dropdown set to "Arc Seconds". Under "Display search results in tabs for:", "Observation Request (AOR)", "Level 2 (PBCD)", and "Level 1 (BCD)" are checked. The "More options" section is expanded, showing "Apply spatial constraints to:" with "AORs" selected. "Select the filter to search by:" has "Instrument Parameters" selected. Under "All IRAC", "IRAC Map", "IRAC Map PC", "3.6 μm", "4.5 μm", "5.8 μm", and "8.0 μm" are checked. Under "All IRS", "IRS Stare", "IRS Map", "IRS Pickup Image", "Hi Res 9.9-19.6 μm", "Hi Res 18.7-37.2 μm", "Low Res 5.2-8.7 μm", "Low Res 7.4-14.5 μm", "Low Res 14.0-21.7 μm", "Low Res 19.5-38.0 μm", "PU Blue 13.3-18.7 μm", and "PU Red 18.5-26.0 μm" are unchecked. Under "All MIPS", "MIPS Phot", "MIPS Scan", "MIPS SED", and "MIPS TP" are unchecked. The "Readout Mode" section has "Full Array", "Stellar Mode", "High Dynamic Range", and "Subarray" checked. At the bottom, there are input fields for "Frame Time Min (seconds)", "Frame Time Max (seconds)", "Start Date (UT)", and "End Date (UT)". A "Search" button is at the bottom right.

Step 3. The Search results

This is what we get for search results. Note that the search panel can be retrieved again at any time by clicking on the blue “Spitzer Searches” tab in the upper left.

Target name	RA (J2000)	Dec (J2000)	NAIF ID	Instrument/Mode	AORKEY	AOR label	Observation start time	Observation end time
CG4	7h33m40.00s	-46d52m00.0s		IRAC Map	16805376	IRAC - CG4	2006-03-27 06:49:28	2006-03-27 07:39:13
cg4	7h33m56.20s	-46d56m23.5s		IRAC Map	24250880	IRAC_cg4	2007-12-29 04:58:41	2007-12-29 07:48:25
PGC0021338	7h35m07.50s	-46d55m42.0s		IRAC Map PC	43725824	CFS-PGC0021338	2012-04-30 21:10:40	2012-04-30 21:16:00
brc48	7h34m24.92s	-46d54m12.4s		IRAC Map	6029824	IRACbrc48	2003-12-19 20:45:27	2003-12-19 20:49:55

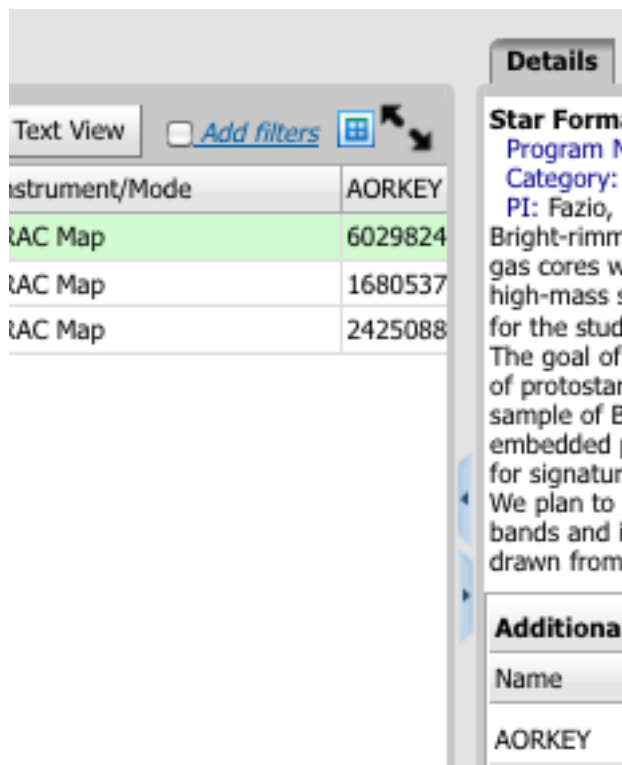
Additional Information

Name	Value
AORKEY	16805376
Readout mode	1
Array 3.6	1
Array 4.5	1
Array 5.8	1
Array 8.0	1

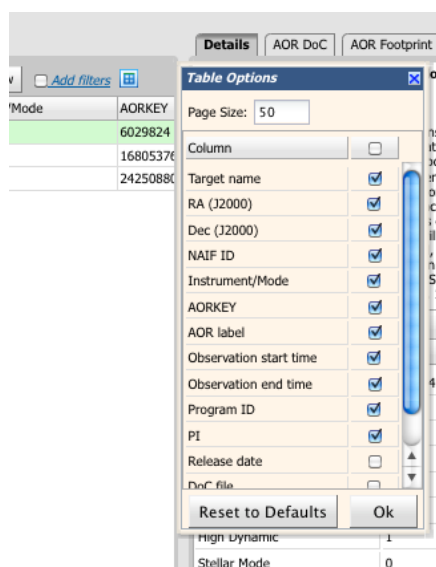
Step 4. Understanding these results -- the AOR tab

OK, so what does this all mean? It comes up by default with the AOR tab in front. This is the most succinct summary of the observations.

For CG4, there are four IRAC observations. On the screen (see figure above), you can see four lines. The columns (some of which you can see in the snapshot above) include target name (which can be cryptic; it was provided by the original observer), the RA/Dec, the NAIF ID (only applies to moving targets, which these are not), the instrument and mode, the AORKEY (a large integer that uniquely identifies the observation within the mission, sort of like a license plate identifies your car uniquely in your state -- some are just as readable), the AOR label (which can be cryptic; it was provided by the original observer), and the observation start/end date/time. On your screen, you can scroll right and see additional information, such as the program id, and the PI for the program. You can grab the slider that divides the two panes and drag it left or right to show more columns; it may or may not be obvious that more information is available at the top of the results pane. This figure shows the slider (blue left-right arrows in the bottom center-right of this figure) as well as the rest of the information from the top of the results window (top center-left, ending with the black arrows):



You can add additional columns or rows to the display. Click on the two wheels icon on the upper right of the search results pane to pull up a popup with a place to select which columns, and how many rows per page, to show.



In this case, for this example, the defaults are ok.

If you scroll over in your results window, you can find the observation start and end times. In this search, I have one observation taken in 2003, centered on a target called "brc48" (AORKEY 6029824), I have another observation taken in 2006, centered on a target called "CG4" (AORKEY 16805376), I have a third observation, taken in 2008, centered on a target called "cg4" (AORKEY 24250880), and I have a

fourth observation, taken in 2012, centered on target called “PGC0021338” (AORKEY 43725824). If you scroll further right in your window, you will see the four PIs from the four different programs.

Other than that, how can you tell the difference between these four observations? Read on ...

Step 5. Understanding these results -- the Details panel and Visualization

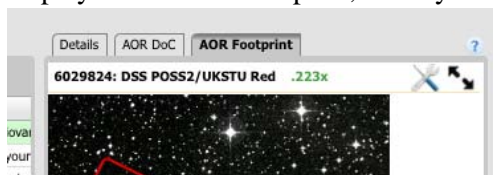
Try clicking on each of the four lines in the AOR window. Note that the contents of the panel on the right, the "Details" pane, changes. The default "Details" tab within the pane gives you background information on the original program and observation parameters. These are important things, but peripheral to our task at hand, so for now, just satisfy yourself that you can see the contents change.

Now, click on the other tab in the Details pane, the "AOR footprint" tab. This is very useful, because it shows you the complete footprint of the AOR in question on a background image. The purpose of this is so that you can compare the region of sky covered by, say, four similar AORs, which is what we need to do here.

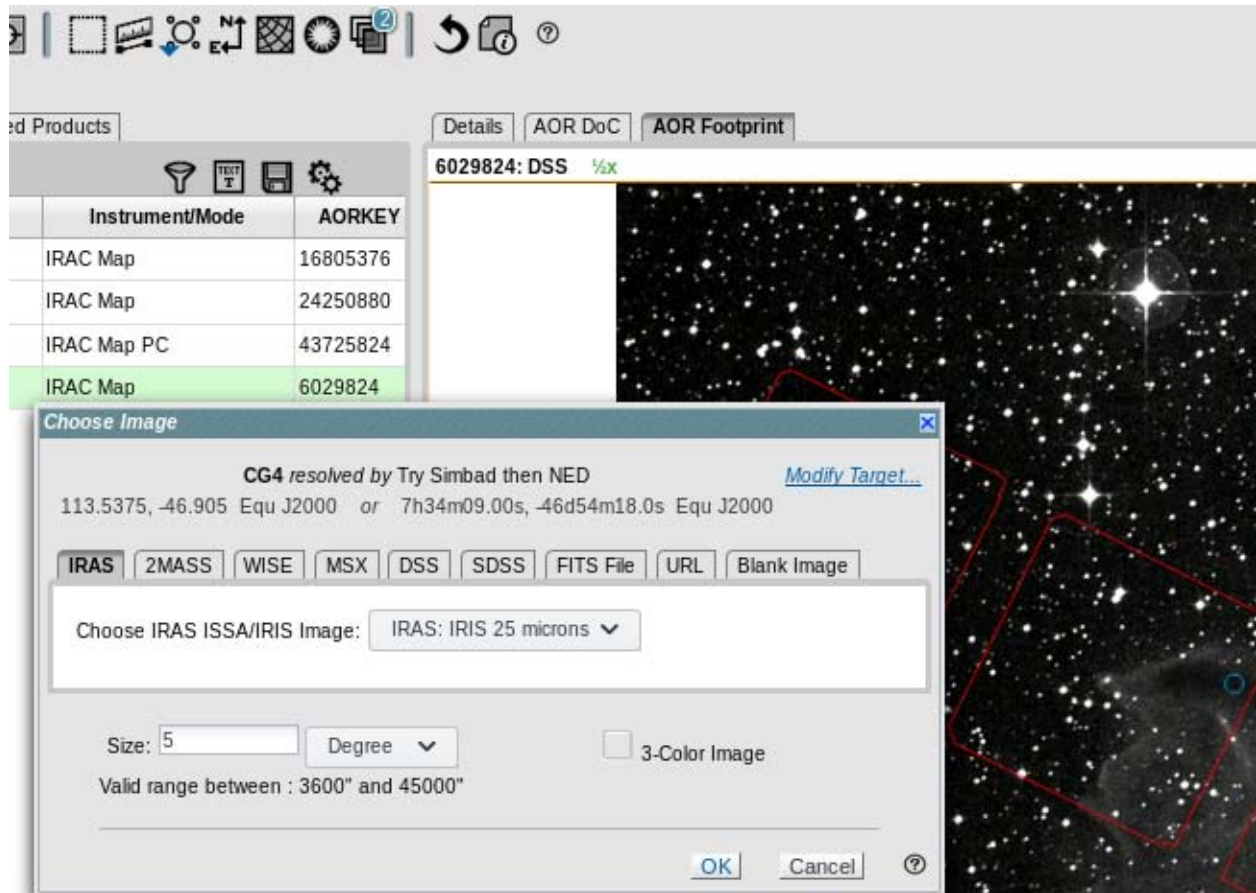
The SHA tries to choose background images intelligently, and generally it does ok. In this case, two of the AORs come up visualized on top of DSS images, and the other two (slightly larger ones) come up on IRAS images. You can see a brief description of the background image on the top of the image, under the tabs. It also has the AORKEY that is being visualized.

With any background image, if you move your mouse over the image, you will get a real-time readout of the position and flux density in a pop-up window.

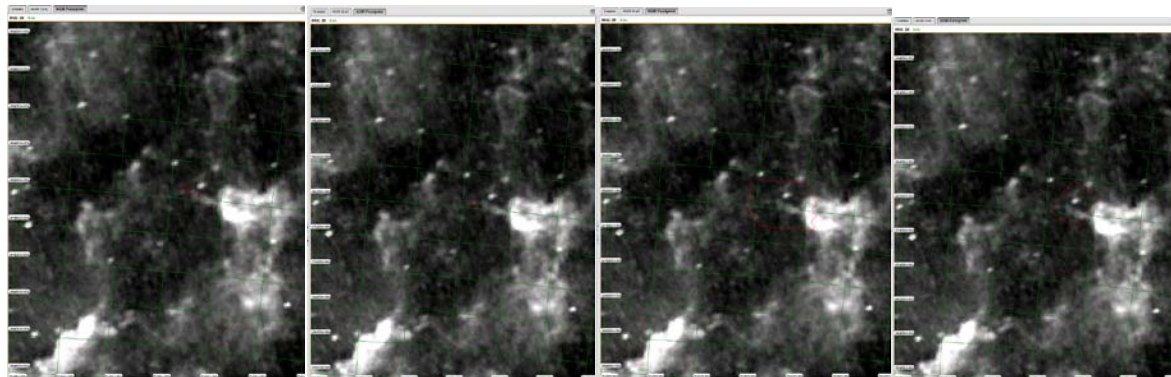
If you want to visualize one on top of a different image (wavelength, size, etc), you can change the background image. Let us change the visualization of the AORKEY 6029824 observation (on the DSS image) to be on an IRAS image. While this observation is highlighted in the list, and with the DSS image displayed under the footprint, move your mouse over the image.



Let your mouse hover on top of any of tool icons on the top pane just above the “Position” text. The tool tip will tell you more about what this icon does. Click on the icon showing a paper sheet with top left corner bent and a plus-sign near the lower right corner (2nd from the left). This enables you to change the background image. It produces a pop-up window. Check to be sure the target is right by checking that it has about the right coordinates. Pick an IRAS/ISSA image at 25 microns. Pick a 5 degree size. Here is what the relevant part of my screen looks like right before I clicked OK; note that you can create a 3-color image for the background if you wish:



Here are the three footprints for the three IRAC AORs (obtained separately and screen-snapshotted here), with a coordinate grid overlaid for comparison (click on the grid-looking icon, about sixth from the right on the icon row, to add this to yours):



The first one, on the left, is the 2012 observation (AORKEY 43725824), the second one is the 2003 observation (AORKEY 6029824), the third one is the 2006 observation (AORKEY 16805376) and the fourth one is the 2007 observation (AORKEY 24250880).

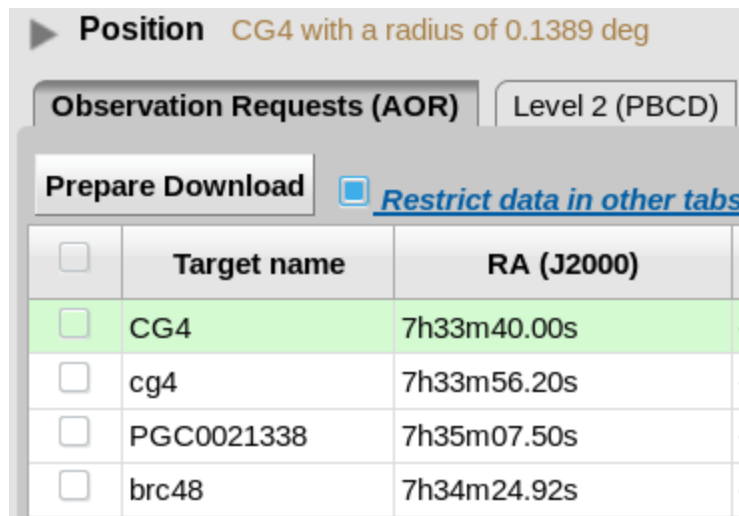
So, we have already made significant progress! The footprints of the 2003 and 2012 observations are very much smaller than the other two observations. We do not want these small observations -- since they have less sky coverage, the chances of them containing a new YSO (our scientific goal for this analysis) are smaller than either of the other two observations.

Now, how can we pick between the other two observations? The one with irregular edges looks like it covers a larger area, so on the face of it, it seems like that is the best one to pick. BUT the fact that it has irregular edges betrays something about how the data were taken (which can also be seen in the Details tab under the observation configuration). And, it turns out, this has an effect on what mosaics are available. Read on!

Step 6. Understanding the results -- the Level 2 (PBCD) tab

OK, so now we know that we are **not** interested in the 2003 and 2012 observations (the ones with "brc" and "PGC" in the target and AOR names, the ones with Fazio and Tully as the PIs, AORKEYs 6029824 and 43725824). But we need to be able to pick between the other two observations. Let us do this by investigating which products are available.

In order to see what Level 2 (PBCD) products are available, check the boxes next to the 2006 and 2007 observations, and click on "Restrict data in other tabs."



The screenshot shows a web interface with a position indicator "CG4 with a radius of 0.1389 deg". Below it are two tabs: "Observation Requests (AOR)" and "Level 2 (PBCD)". Under the "Observation Requests (AOR)" tab, there is a "Prepare Download" button and a checkbox labeled "Restrict data in other tabs" which is currently unchecked. Below this is a table with two columns: "Target name" and "RA (J2000)". The table contains four rows of data, with the first row highlighted in green.

<input type="checkbox"/>	Target name	RA (J2000)
<input checked="" type="checkbox"/>	CG4	7h33m40.00s
<input type="checkbox"/>	cg4	7h33m56.20s
<input type="checkbox"/>	PGC0021338	7h35m07.50s
<input type="checkbox"/>	brc48	7h34m24.92s

What this does is filter the information displayed in the other tabs, e.g., "In the other tabs I have here, show me only the many files associated with these two AORs, not all four." By default, it shows you everything in the other tabs. Similarly, if, instead of two AORs, you want to see the files associated with just one of the AORs (say, the one from 2006), just leave that one AOR clicked.

Then, click on the "Level 2 (PBCD)" tab.

AORKEY	Instrument/Mode	Bandpass	RA (J2000)	Dec (J2000)	File typ
24250880	IRAC Map	IRAC 3.6um	7h35m6.83s	-46d49m41.5s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m29.33s	-46d50m13.8s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m33.57s	-47d3m47.9s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m31.55s	-46d58m18.5s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m43.54s	-46d58m25.1s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m43.49s	-46d49m33.2s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m44.46s	-46d45m41.3s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m7.74s	-46d58m31.4s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m55.47s	-46d46m0.0s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m44.98s	-46d53m59.8s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m54.98s	-46d49m7.3s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m18.23s	-46d54m17.9s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m57.67s	-47d3m14.2s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m21.65s	-46d45m13.9s	Ima
24250880	IRAC Map	IRAC 4.5um	7h33m16.96s	-47d3m13.1s	Ima

Note that there are a TON of lines here. From 2 AORs, we have gone to 188 lines (see the top center of the search results pane), and there will be more files that come with these data when you download them. That is why it is best to start with the AOR tab -- it's a higher-level summary of the observations.

OK, now it gets a little more complicated. For the most generic case, for most observations you will find in the SHA, you will get results like for the 2006 observation; the 2007 observation was taken in a different, rarer way. This is a good example, because it shows how to use the SHA to investigate the products. In all likelihood, your search of any given patch of sky will be simpler!

Look at the Level 2 (PBCD) results. The first text column is the AORKEY. Click on the AORKEY column label to sort by this column, in increasing order.

AORKEY	Instrument/Mode	Bandpass	RA (J2000)	Dec (J2000)	File typ
16805376	IRAC Map	IRAC 4.5um	7h33m40.18s	-46d52m1.2s	Ima
16805376	IRAC Map	IRAC 3.6um	7h33m40.18s	-46d52m1.2s	Ima
16805376	IRAC Map	IRAC 5.8um	7h33m40.18s	-46d52m1.2s	Ima
16805376	IRAC Map	IRAC 8.0um	7h33m40.18s	-46d52m1.2s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m6.83s	-46d49m41.5s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m29.33s	-46d50m13.8s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m33.57s	-47d3m47.9s	Ima
24250880	IRAC Map	IRAC 3.6um	7h35m31.55s	-46d58m18.5s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m43.54s	-46d58m25.1s	Ima
24250880	IRAC Map	IRAC 3.6um	7h34m43.49s	-46d49m33.2s	Ima

Look at how few files are listed for 16805376 (which is the 2006 observation), and how many for 24250880 (which is the 2007 observation). This is your first clue that maybe you want to use the data for the first one and not the second, even though it is covering a smaller area of sky.

The next column is the instrument and mode. These are all IRAC Mapping observations.

The next column is the bandpass. Look at the four files for 16805376 (which is the 2006 observation) -- there is one per bandpass. All the rest of the files in here (184-4 pages) are from 24250880. Ugh! To confirm that these are all images, look at the 6th text column – they are all images.

To see what the data look like, pick one of these (*maic.fits) files from 16805376 and go over to the "Details" panel. Click on the "Data" tab. That is the real mosaic that is shown there. You can scroll through the 4 mosaics (select different rows in the search results), and even rotate North up if you want (find the “rotate North up” icon). You can also still see some of the image artifacts that come from the bright stars that are incompletely removed!

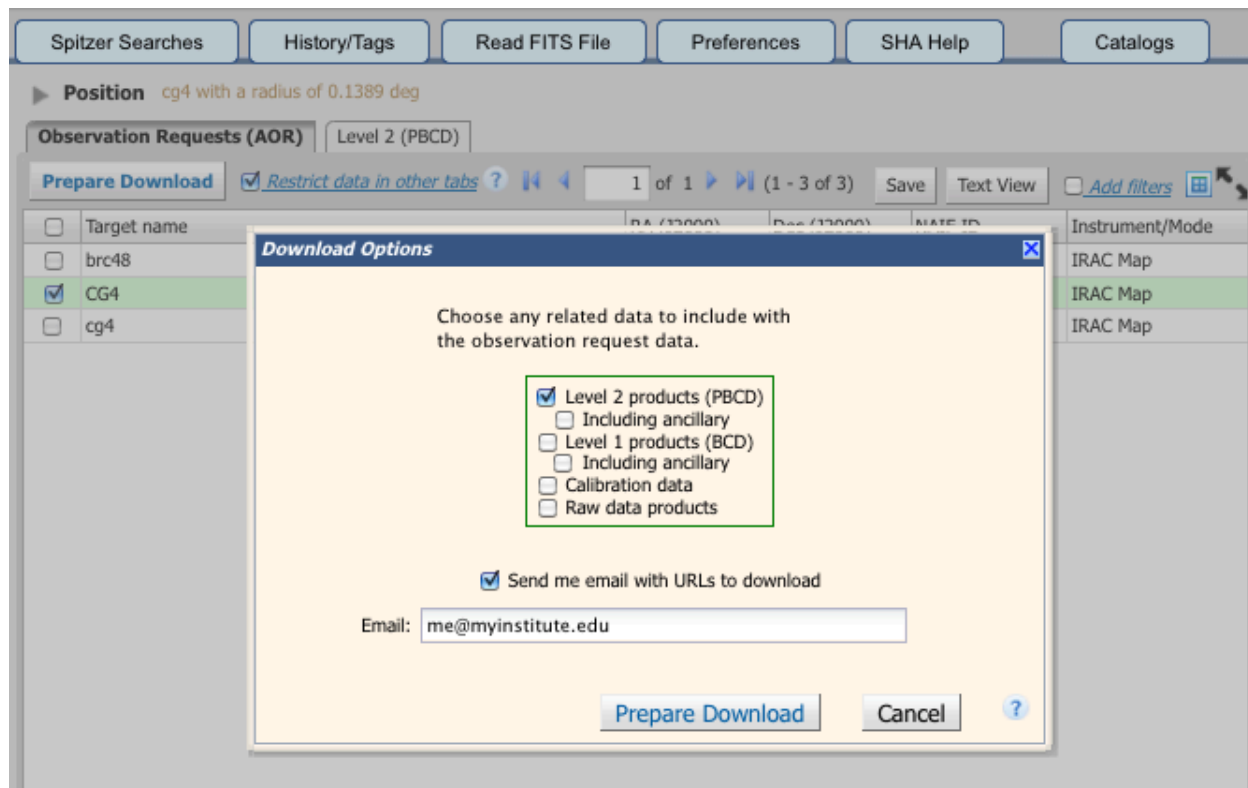
Why are there so many more files for 24250880 rather than 16805376?

What is going on is that the data were taken in different ways. The observation from 2006 (AORKEY 16805376) is a simple map. As such, our pipelines understand exactly how to handle that and make a single mosaic for the whole observation. This is what you want to download for the lowest-energy solution. Proceed to the next step on downloading the data. The observation from 2007 (AORKEY 24250880) was taken using "cluster targets", which means that it covered an irregularly-shaped patch of sky where the observations were (among other things) designed to cover about the same area, regardless of spacecraft orientation. BUT, the Spitzer pipeline is *a priori* uncertain if these cluster targets are designed to cover one contiguous patch of sky, or many different tiny patches all over a 1.5 degree diameter circle. So, what the pipeline does is create one little mosaic for each pointing. If you want to create your own mosaic from the Level 1 (BCD) data, this distinction does not matter. If you are interested in a quick look of a complete mosaic, then you want the simplest solution. And, the SSC pipelines are pretty good – our pipeline mosaics are fairly nice, and may be sufficient for your science (see the corresponding Instrument Handbook for more information). For this example, take the low-energy solution and pick the simpler mosaic -- the observation from 2006 (AORKEY 16805376).

OK, go back and look at the Level 2 (PBCD) screen. Scroll right. You will see the long filenames for these observations. Note that the four files associated with AORKEY 16805376 all have that number in the filename itself, and that they are all listed as *maic.fits. This means that they are a MosAIC. You will get other files, such as uncertainties, when you download the data.

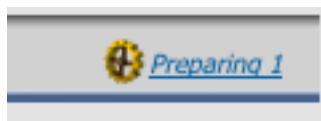
Step 7. Downloading the data

So, now we have picked the AOR from 2006, AORKEY 16805376. Go back to the AOR window, check just that AOR (uncheck both of the other AORs), and click on "Prepare Download" right above the column full of the check boxes. You get a popup asking what data you want to download. Pick the Level 2 products. If you want to be sure you have everything associated with this AOR, pick the ancillary data too.

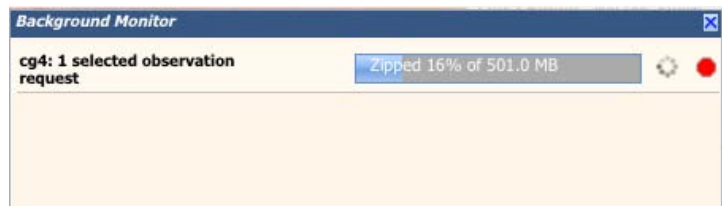


Click on "prepare download." Enter an email if you want to be notified when the packaging is complete and it is ready for download

You get a new panel then, a "background monitor" which first looks like this in the upper right of your screen as it packages your data:



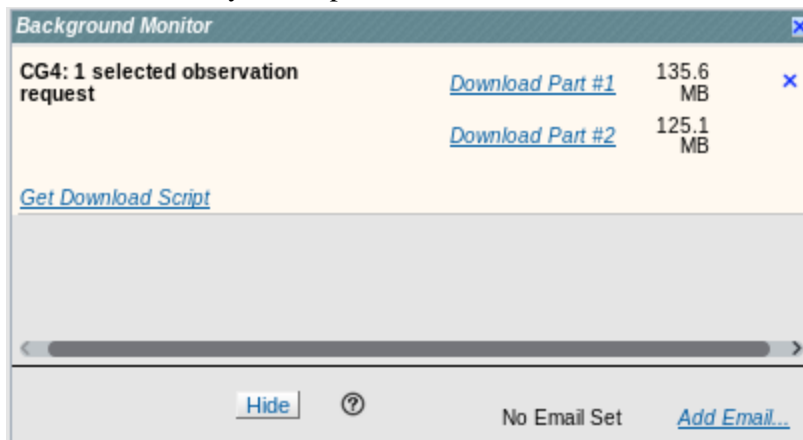
If you click on the link quickly, before it is done, you can see it package:



then like this when it is done, if you have not opened it:



and like this when you do open it:



Click on "Download now" in order to download your data.

The file will be a ".zip" file, and either your browser will be configured to ask you what you want to do with the file, or it may automatically unzip it and put the files somewhere you have configured (possibly a "Downloads" folder). This is entirely controlled locally by your browser, and each one is configurable and thus different on each computer. Find this file. You may need to search for "recently modified files" or something similar.

Step 8. Unpacking the data

If your computer did not by default unpack it, you will need to unzip the file. You can double-click, or from a Mac terminal window, type "unzip <blah>.zip" where <blah> means whatever the file is called. For this example, it was called "CG4-1-selected_AORs.zip".

For this example, we do "unzip CG4-1-selected_AORs.zip" from a terminal window, and this is what happens:

```
user% unzip CG4-1-selected_AORs.zip
Archive:  CG4-1-selected_AORs.zip
Files 1-13
inflating: r16805376/ch4/pbcd/SPITZER_I4_16805376_0000_3_E4643337_maic.fits
inflating: r16805376/ch1/pbcd/SPITZER_I1_16805376_0000_3_E4643376_maic.fits
inflating: r16805376/ch3/pbcd/SPITZER_I3_16805376_0000_3_E4643451_maic.fits
inflating: r16805376/ch2/pbcd/SPITZER_I2_16805376_0000_3_E4643768_maic.fits
inflating: r16805376/ch4/pbcd/SPITZER_I4_16805376_0000_3_A23926000_munc.fits
inflating: r16805376/ch4/pbcd/SPITZER_I4_16805376_0000_3_A23926001_mcov.fits
inflating: r16805376/ch1/pbcd/SPITZER_I1_16805376_0000_3_A23926514_munc.fits
inflating: r16805376/ch1/pbcd/SPITZER_I1_16805376_0000_3_A23926515_mcov.fits
inflating: r16805376/ch3/pbcd/SPITZER_I3_16805376_0000_3_A23927127_munc.fits
inflating: r16805376/ch3/pbcd/SPITZER_I3_16805376_0000_3_A23927128_mcov.fits
inflating: r16805376/ch2/pbcd/SPITZER_I2_16805376_0000_3_A23929213_munc.fits
inflating: r16805376/ch2/pbcd/SPITZER_I2_16805376_0000_3_A23929214_mcov.fits
```

```
inflating: r16805376/QualityAnalysis_16805376.README
inflating: README.txt
```

If you double-click to unzip, you may not get this readout, but all these files will (should) appear in whatever directory you have placed the zip file. Note that they are embedded within directories. These directories may seem arcane, and somewhat silly for these few files, but this directory structure makes a lot more sense when you have, say, 200+ files to download.

Step 9. What are all these files?

The files that you most likely care about the most are the `*maic.fits` files. These are the mosaic files. There is one per channel. The subdirectory "ch1" means the same as the "I1" in the filename -- it means IRAC channel 1, also known as IRAC-1, a.k.a. 3.6 microns. Same for ch2=I2=4.5 μm , ch3=I3=5.8 μm , and ch4=I4=8 μm . If you are trying to interpret these same sorts of search results for MIPS, MIPS-1=ch1=24 μm , MIPS-2=ch2=70 μm , MIPS-3=Ch3=160 μm . You may wish to move and rename these files straightaway into something more human-readable.

The other files are also coded by channel, but they are uncertainties ("munc", or Mosaic UNCertainties) and coverage ("mcoV", or Mosaic COVerge). Coverage matters, because it tells you how many frames per position you have. The coverage map (and, more indirectly, the uncertainties) reflect how many times each tiny patch of sky was observed.

You have IRAC mosaics of the CG4 region! Now, try obtaining some MIPS-24 data for this region

Recipe 3. MOPEX: Mosaic IRAC S-COSMOS Data

This is a recipe for reducing medium deep (~15 minute integration with more than 10 exposures), blank field IRAC survey data in all four bands, similar to the SCOSMOS Legacy Science Program survey (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/spitzermission/observingprograms/legacy/>). In this recipe, we outline the steps needed to use MOPEX to produce science grade images from IRAC basic calibrated data (BCD). We focus on the detailed set-up for channel 1, and then outline what settings need to be changed for the other three IRAC channels.

Background subtraction for this type of data and object density is optimally treated with the default "Overlap" pipeline. Outlier rejection is best treated with a box outlier and temporal outlier rejection. The more involved "Dual Outlier" method is not needed with this level of coverage. Images are interpolated onto a final grid of 0.6" per pixel using the "Drizzle" algorithm. This optimally samples the PSF by taking advantage of sub-pixel offsets in the data.

There are additional recipes in this cookbook for other types of IRAC data, including deeper and shallower blank field data. You should compare the settings to select those optimal for reducing your data. However, this recipe is the most detailed, and should be read first before moving on to the others.

Before starting, it is a good idea to browse through the IRAC Instrument Handbook (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/iracinstrumenthandbook/>). If you are familiar with MOPEX you can simply load the provided namelist for channel 1 and modify it (http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/irac_medium.nl).

3.1 Requirements

You need to have MOPEX (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/tools/mopex/>) installed to follow the instructions in this recipe.

If your data have been processed by version S18.0 or later of the IRAC pipeline, you do not need to run the pre-processing steps because corrected BCD (CBCD) data will be provided. However, if you do need to run the pre-processing, IDL version 6.2 or newer and the IDL Astronomy Users's Library (<http://idlastro.gsfc.nasa.gov/>) are required.

3.2 Organizing Files and Directories

Before starting with this recipe, you should download the example data set using the Spitzer archive interface. The COSMOS IRAC AORs used in this recipe are 15536640, 15537152, 15540224, 15540736, 15541504, 15546112, 15546368, 15547392, 15547904, 15552512, and 15553024. These total ~6GB of data for all four channels. Specify that you need the Basic Calibrated Data (BCD) and the corrected BCD

(CBCD) for all four IRAC bands. The BCDs are single images which have been run through the Spitzer online calibration pipeline and have had some basic artifact clean-ups applied (orientation flipping, dark subtraction, flat field correction, conversion into units of MJy/sr).

You likely have several AORs, so multiple directories will be created, each with a subdirectory labeled *ch1/*, *ch2/*, *ch3/*, and *ch4/*. Under each of the channel directories you will find a *bcd/* directory containing the calibrated data. We strongly encourage you to download the latest processing of data from the Spitzer Heritage Archive instead of using earlier processings.

You need to make several lists of input files for MOPEX. A separate list should be made for each IRAC channel. The only required list contains the input images (*_bcd.fits* or *_cbcd.fits* files). However, a list of the uncertainty maps (*_bunc.fits* or *_cbunc.fits* files) and a list of the bad pixel masks (*_bimsk.fits* files) is strongly recommended if you wish to remove artifacts. Give the following command in the directory under which you have downloaded the data from the 11 AORs mentioned above:

```
unix% find . -name "*.fits" > files.lst
```

Then give the following commands:

```
unix% grep ch1 < files.lst | grep _cbcd.fits > ch1_bcd.lst
unix% grep ch1 < files.lst | grep _cbunc.fits > ch1_bunc.lst
unix% grep ch1 < files.lst | grep _bimsk.fits > ch1_bimsk.lst
unix% grep ch2 < files.lst | grep _cbcd.fits > ch2_bcd.lst
unix% grep ch2 < files.lst | grep _cbunc.fits > ch2_bunc.lst
unix% grep ch2 < files.lst | grep _bimsk.fits > ch2_bimsk.lst
unix% grep ch3 < files.lst | grep _cbcd.fits > ch3_bcd.lst
unix% grep ch3 < files.lst | grep _cbunc.fits > ch3_bunc.lst
unix% grep ch3 < files.lst | grep _bimsk.fits > ch3_bimsk.lst
unix% grep ch4 < files.lst | grep _cbcd.fits > ch4_bcd.lst
unix% grep ch4 < files.lst | grep _cbunc.fits > ch4_bunc.lst
unix% grep ch4 < files.lst | grep _bimsk.fits > ch4_bimsk.lst
```

You need to edit the lists to remove all short frames, i.e., frames with “0000_0000” and “0000_0001” in their file names. E.g., the following command should work:

```
sed -e '/0000[_]0000/ {d; b} /0001[_]0000/d' ch1_bcd.lst > ch1_bcd.list
```

Finally, you need to download the bad pixel maps (IRAC Pmask files):

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/>.

3.3 MOPEX Overview

The MOPEX tool processes calibrated imaging data (BCDs and CBCDs) into a science grade mosaic and can produce a photometric catalog. MOPEX includes modules that match the background, astrometrically

register and re-sample the images, remove outliers, combine the individual data into a final mosaic, and measure photometry. MOPEX can be run on the command line or with a graphical interface (GUI). This recipe is written for the GUI interface, but the provided setup files can be used with the command line version of MOPEX.

The following assumes you are starting from scratch. If you are familiar with MOPEX you can simply load the provided namelist file and modify it:

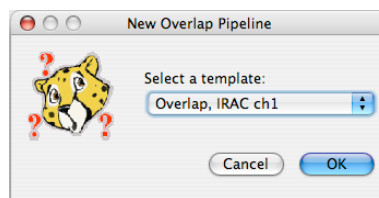
(http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/irac_medium.n).

MOPEX has several major components that can operate together as one pipeline or separately on different sets of data. The major components are "Overlap", "Mosaic", "APEX multi-frame", and "APEX single frame". Overlap is designed to remove image background variations due to foreground light sources. Mosaic removes defects and spurious pixels (e.g. cosmic rays), reassembles the data onto a common pixel grid, and combines your data into a science grade mosaic with a corresponding noise map. The two APEX modules find objects and measure photometry on the final calibrated data. APEX multi-frame measures photometry on each frame individually, while APEX single frame measures it on the combined mosaic.

This recipe will detail the set-up of a pipeline and explain the various parameter choices. All of the settings and how to change them will be explained in detail for IRAC Channel 1 (ch1). After the channel 1 example, the setting changes required for the other three channels will be explained.

Note that when a sub-menu (including the help menu, but not including the component menu) is open in the MOPEX GUI, all other windows are frozen. If you can not modify a setting or scroll through the help menu, check for other open windows. This is a feature of JAVA and can not be easily corrected.

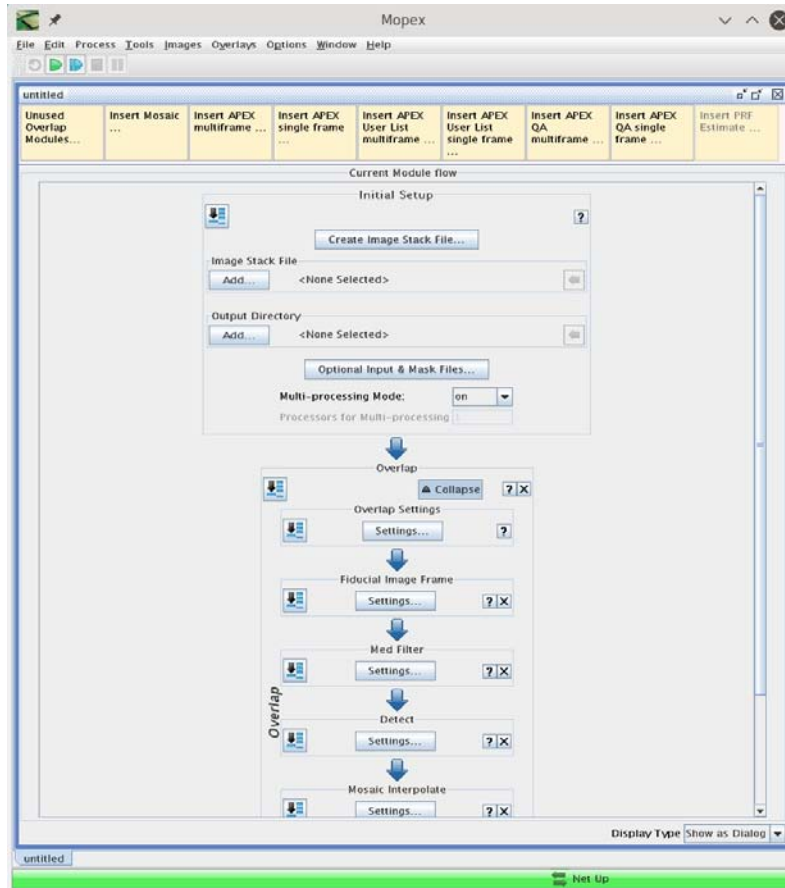
To begin, start the MOPEX GUI and select "New Overlap Pipeline" from the file menu. This will bring up a dialog box asking which template you would like to use:



Select "Overlap, IRAC ch1", and click OK. This will bring up a window with the default IRAC ch1 overlap pipeline and a second window listing the modules you can add to the overlap workflow.

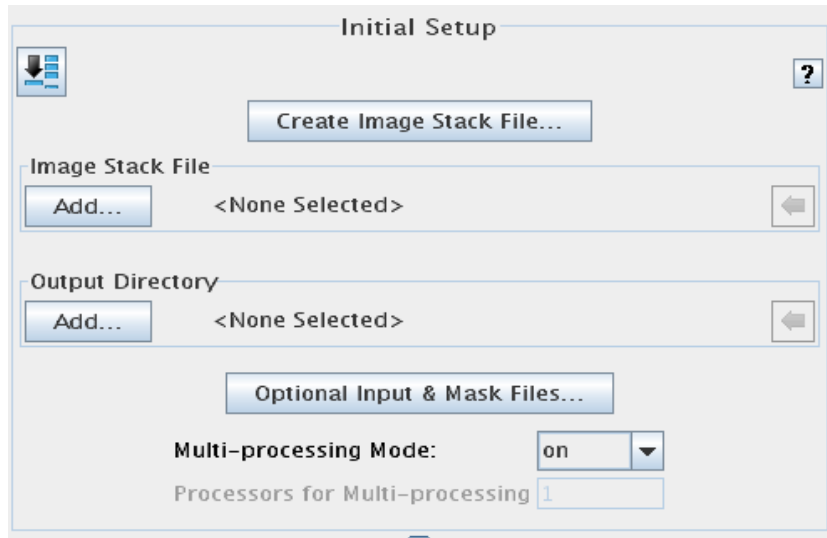
The GUI is laid out as a data flow diagram. The first box specifies the input data and each subsequent box represents a module in the data reduction pipeline. Each module acts on the data from the previous module and then passes the data to the next module in the order specified by the GUI window. When you add a module it will be placed in the appropriate position in the work flow. If the module you added depends on previous modules, you will be informed which modules you need to add.

You can insert other components to the pipeline by clicking on the buttons along the top of the window. When you insert other components, additional windows will open, listing modules in each component's workflow. When you want to run the pipeline you can click on the green play button at the top of the window.



3.4 Initial Setup

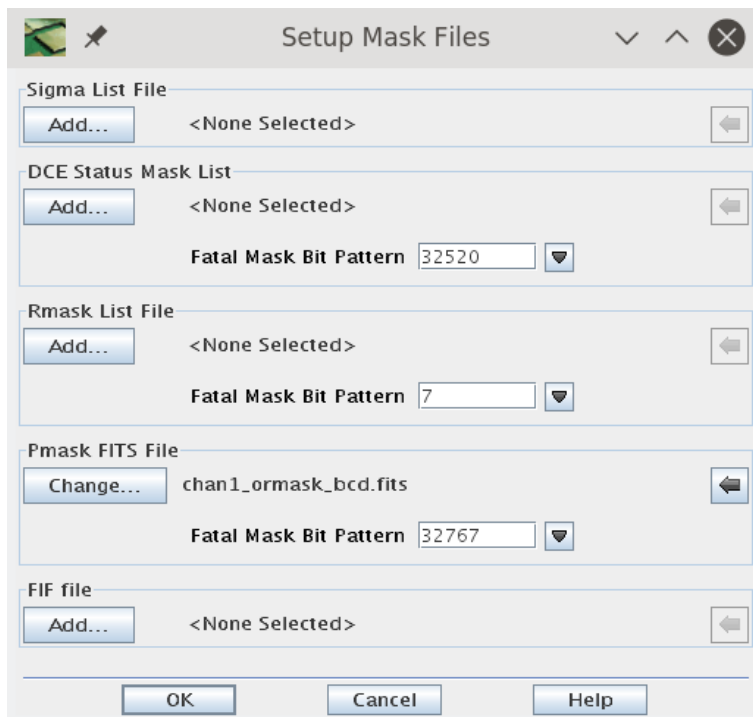
The first step is to specify which data you want reduced and what you want the final image to look like. In the initial setup box, specify which file lists MOPEX should use for the stacking. The first dialog box should look like this:



Click "Add..." under "Image Stack File" and select the file containing the list of IRAC Channel 1 image files you created earlier.

Click "Add..." under "Output Directory" and specify where the intermediate and final data products should be saved.

Next, click "Optional Input & Mask Files..." which will open this dialog box:



In this dialog box you specify the input uncertainty and mask files, along with which flags will be used to reject pixels (the Fatal Mask Bit Patterns). The complete list of bad pixel flags can be found in the IRAC

Instrument Handbook section 7.1.1. Clicking on the downward facing arrow to the right of each "Mask Bit Pattern" window will open a window which allows you to turn each bit on or off.

Click "Add..." under "Sigma List File" and specify the list of cbunc files created earlier.

Click "Add..." under "DCE Status Mask List" and specify the list of bad pixel (bimsk) files created earlier.

Use the default Fatal Mask Bit Pattern 32520 (bits 3, 8-14).

Rmasks contain a list of pixels rejected by previous runs of MOPEX. In version 16.3.7 and later of MOPEX there is no need for an input list in most cases. In previous versions a list was needed in the second pass reduction if you were using Drizzle.

Even if you are not providing a list of rmasks you need to set the "Fatal Mask Bit Pattern" for later in the pipeline. For this type of data set the Rmask Fatal Bit Pattern to 10 (use box and temporal outlier) rather than the default 7 (use dual outlier).

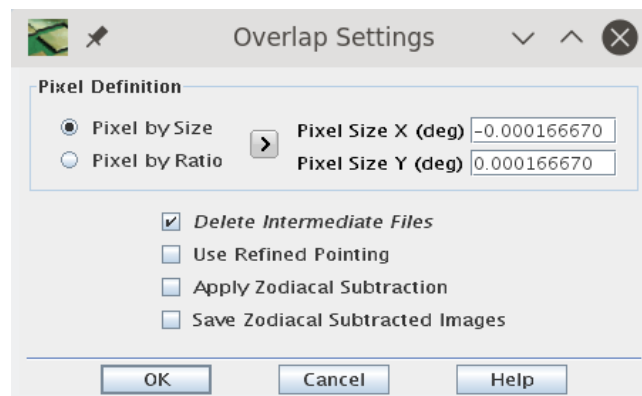
Click "Add..." under "Pmask FITS File" and specify the bad pixel mask you downloaded. For the COSMOS Cycle-2 data the file is "*dec05_ch1_bcd_pmask.fits*". The default "Fatal Mask Bit Pattern" of 32767 is appropriate.

The FIF file records the "Fiducial Image Frame". This file will contain WCS information that allows you to match various sets of Spitzer data to one another and specify the image frame for other channels. Leave this as the default unless you have a FIF you want matched. See the [Helpdesk](#) Knowledgebase for more information about changing the FIF

(<http://sohelp2.ipac.caltech.edu/support/index.php?Knowledgebase/Article/View/161>).

Once you are finished specifying the input parameters click "OK" to return to the pipeline view.

In the pipeline view click on "Settings" under "Overlap Settings". That will bring you to this window:

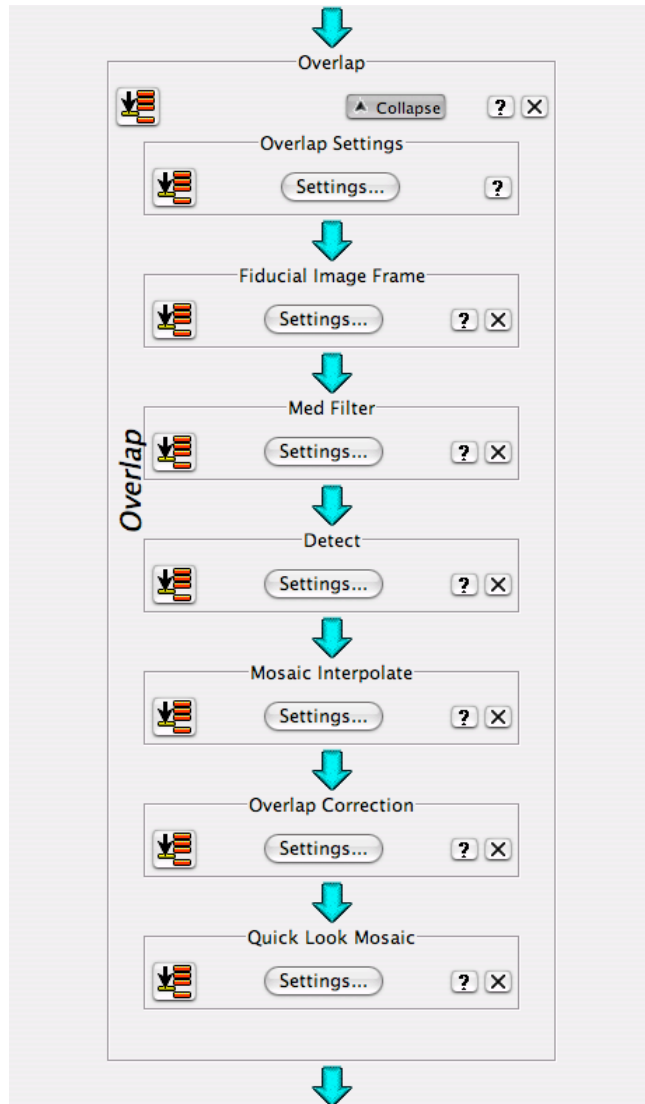


where you can specify the output pixel size. A scale of 0.6 arcseconds per pixel has been found to be optimal. That corresponds to an X pixel size of -0.000166670 and a Y pixel size of 0.000166670. Setting the X pixel size to a negative value will generate an image with RA running to the East, which is the default for most FITS images.

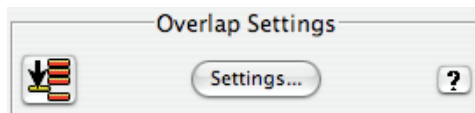
Click OK when you have specified the pixel size to return to the pipeline view.

3.5 Channel 1 Overlap

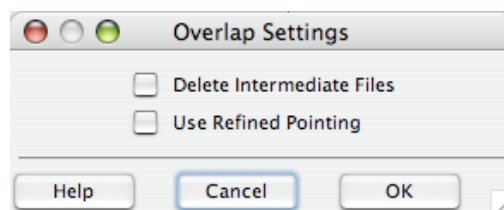
The Overlap module corrects for variable background levels in your data by adding a constant value to every frame. As the name suggests, the additive offset is determined by measuring the background level in overlapping frames. This requires generating a first pass mosaic and masking objects which may contribute to the background. The default settings are optimal. However, if you have very extended objects, a crowded field, or extended emission in your field you may want to adjust some parameters. Below is an example of the default Overlap pipeline:



3.5.1 *Overlap Settings*



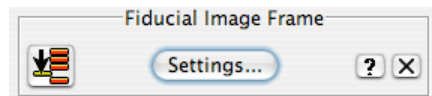
This configures the basic settings for the overlap module. To configure it click on "Settings..." in the "Overlap Settings" box.



You may wish to check the "Delete Intermediate Files" box if you are short on disk space. This will remove the intermediate data products when they are no longer needed.

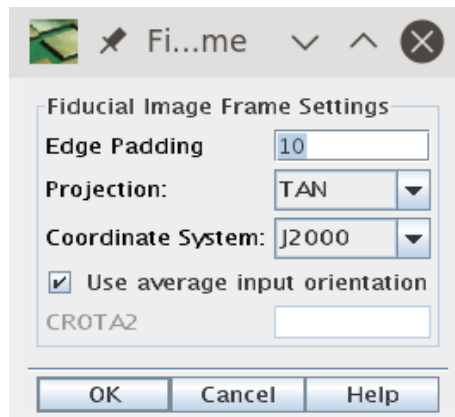
Check "Use Refined Pointing" to use pointing updates if you have applied them to the BCD files. This is generally not required for data taken after GO proposal Cycle 2. The pointing refinement involves identifying point sources in the IRAC frames and correlating them with 2MASS source positions. The pointing refinement typically improves the positional error to less than 0.3 arcseconds and removes any systematic offsets. For more information, see Section 5.3.1 of the IRAC Instrument Handbook.

3.5.2 *Fiducial Image Frame*



The "Fiducial Image Frame" sets the parameters for the overlap mosaic. These are distinct from the final data mosaic. There is no reason to change these settings since this mosaic is only intended for a quick look.

To configure it, click on "Settings..." in the "Fiducial Image Frame" box. This will bring up the following dialog box:



Edge padding specifies the number of blank pixels around the edge of the data. A blank area of this width will be placed on all sides of your final mosaic. The default value of 10 is fine for most cases.

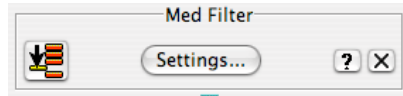
Projection specifies the WCS projection. The default "TAN" is used for most astronomical FITS images. Coordinate System specifies the celestial system for the WCS. Options are J2000, Galactic or Ecliptic. For extragalactic studies J2000 is the appropriate setting.

The "Use average input orientation box" specifies how the WCS rotation is specified. Leaving this box checked will minimize the size of the final fits image by making the final mosaic as rectangular as

possible. However, it is often desirable to specify a fixed rotation to match other data. For COSMOS, uncheck this box and set CROTA2 to 0.00.

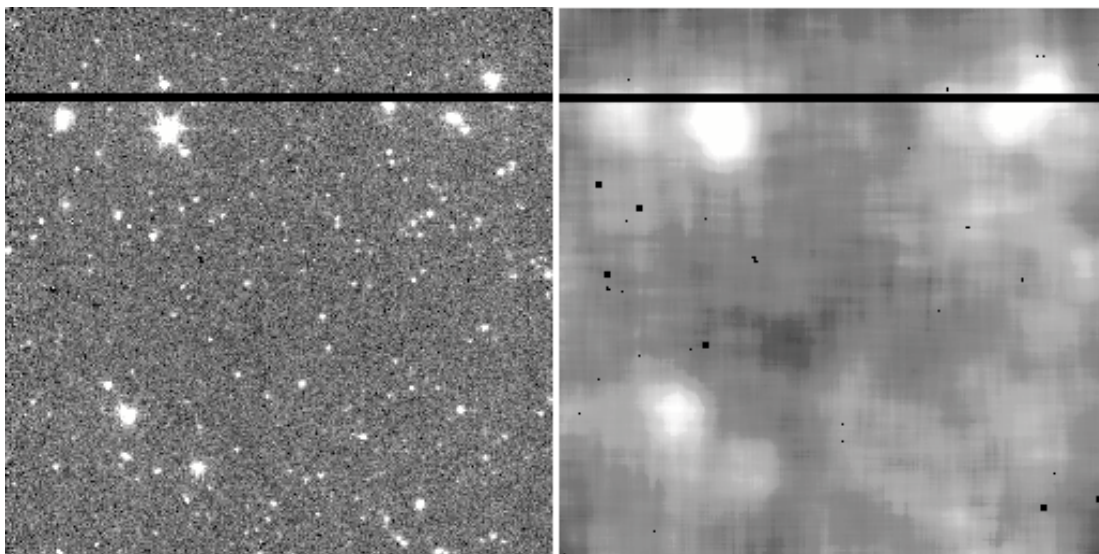
When you are finished, click "OK".

3.5.3 *Med Filter*



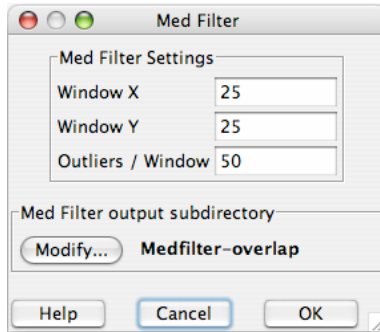
This module filters and removes the background in individual images. It is needed by the "Detect" module to find objects. Med filter constructs a background by measuring a clipped median in a window around each pixel and subtracting it. If you have very extended sources or a crowded field, you may want to increase the size of the window and/or increase the number of pixels rejected in each window.

An example image and the background determined by "Med Filter" is shown below.



Note the background around bright extended objects is too high. However this is not important as long as the majority of real objects are found and masked in the next step.

You can control the window size and clipping by clicking on "Settings...", which will bring up this window:

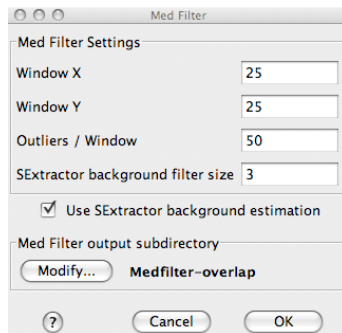


Window X and window Y set the x and y size of the median window. However, this step is computationally intensive and the required time scales geometrically. So increasing the window size by a factor of two will increase the required compute time four fold.

Outliers/Window specifies how many outlier pixels will be removed before calculating the median.

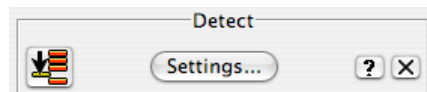
The "Med Filter output subdirectory" specifies where the resulting images will be stored for you to inspect.

In MOPEX 18.2.0, an additional option was added "Use SExtractor background estimation":



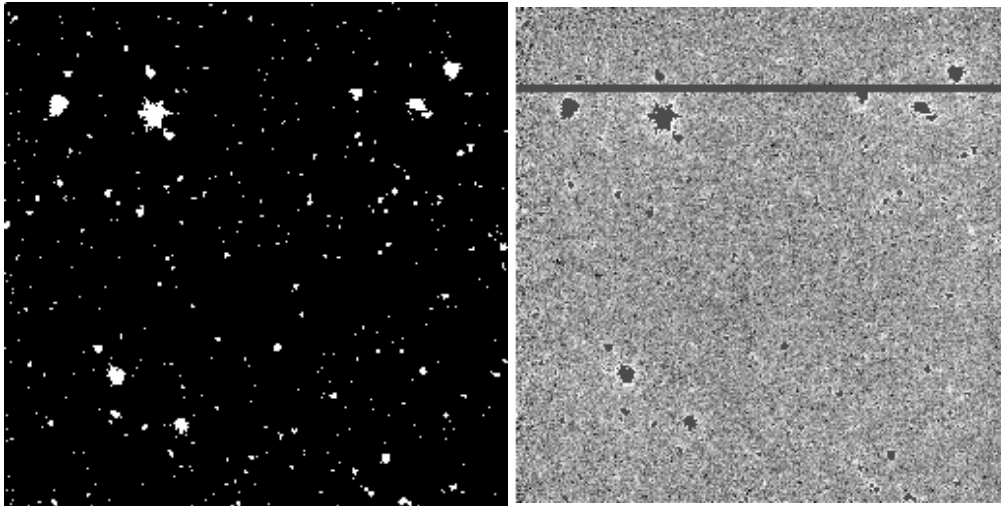
Turning on this flag improves the pipeline speed by about a factor of 2 without affecting the results in any significant way.

3.5.4 Detect



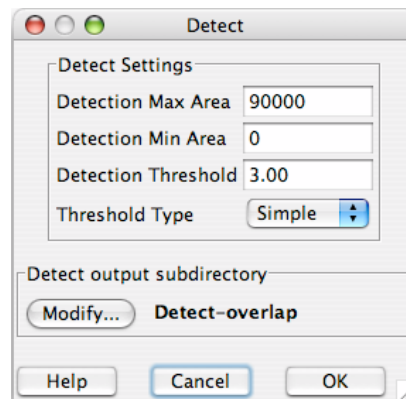
This module detects objects so they can be masked when determining the background in overlapping regions. If you have very extended sources or a crowded field you may want to modify the detection settings to optimally mask the objects.

A detection map from the overlap "Detect" module and a raw image with objects masked is shown below.



One could try to improve the masking by setting the "Detection Threshold" lower. However, as long as the overlap between frames is large and the majority of objects are masked, this is unnecessary.

To configure this module, click on "Settings..." in the "Detect" box.



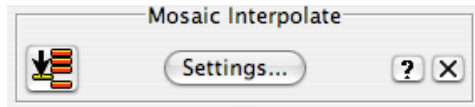
The detection area settings control the maximum and minimum size of an object considered to be real.

The detection threshold is the minimum S/N of the object peak. The noise is determined by the uncertainty maps from the BCD pipeline unless you add a S/N estimator to the pipeline. You should only do that if you feel the provided uncertainty maps are wrong for some reason.

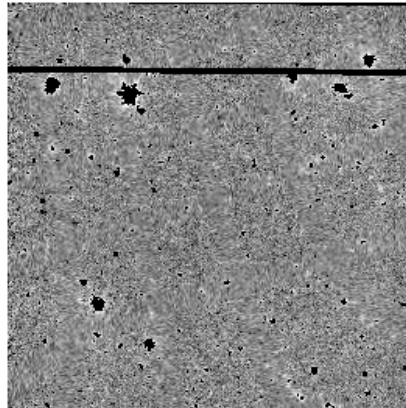
Threshold type determines how objects are de-blended, which is outlined in the image segmentation manual: http://irsa.ipac.caltech.edu/data/SPITZER/docs/files/spitzer/image_segmentation1.pdf.

The "Detect output subdirectory" will contain the output object mask for your inspection.

3.5.5 Mosaic Interpolation

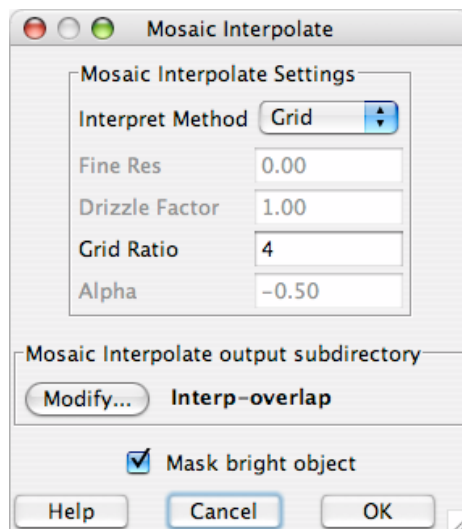


The "Mosaic Interpolate" module does the first pass interpolation of the data onto the FIF. For overlap, this is intended to be a quick look which simply matches the frames to one another so the relative backgrounds can be determined. Therefore, the fastest interpolation, "Grid" (nearest neighbor), is used. If you want to use the quick look mosaic for something you may want to use a better, but slower interpolation than grid. An interpolated image is shown below.



Note the field geometry has changed slightly and there is slight aliasing across the image. The aliasing is due to the fast "Grid" interpolation.

To configure this module, click on "Settings..." in the "Mosaic Interpolate" box.



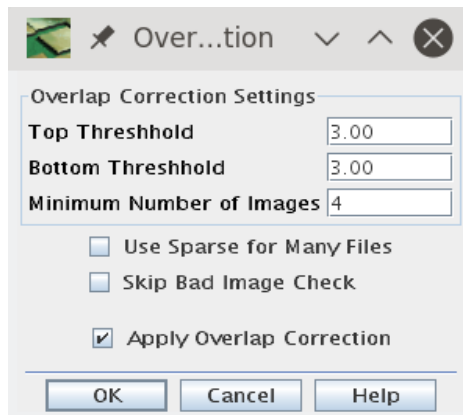
The "Interpret Method" menu sets the interpolation method.

Grid Ratio specifies how fine of a sub-pixel grid will be used for the interpolation.

The "Mosaic Interpolate output subdirectory" specifies the location of temporary files.

3.5.6 *Overlap Correction*

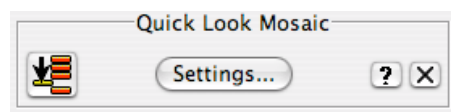
This module calculates and applies a correction which produces a consistent background across your image. If you are having problems matching the background you may want to adjust the outlier rejection in this module by clicking on "Settings...".



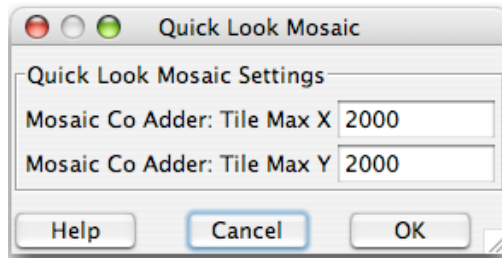
The "Top Threshold" and "Bottom Threshold" and "Minimum Number of Images" settings are used to control outlier rejection. If a frame is "Top Threshold" sigma above or "Bottom Threshold" sigma below the median correction at this position in the mosaic it will be rejected. The "Minimum Number of Images" setting specifies the minimum number of overlapping images at a given position required to do outlier rejection.

If the "Apply Overlap Correction" box is not checked, the determined background corrections will not be applied. They will still be recorded in a table file.

3.5.7 *Quick Look Mosaic*



Note that this module has to be manually added to the flow. This module generates a quick look mosaic with no outlier rejection. This is useful to determine if the background has been properly subtracted. If you already know your overlap settings are optimal and do not want a quick look you can remove this step to increase the pipeline speed by clicking on the "X" at the right of the box. If you have very little memory, or know the size of your final mosaic you may want to adjust the size of the sub-mosaics created by this step by clicking on "Settings..."



The Tile Max X and Tile Max Y settings determine the maximum size of each "tile" made by the mosaic module. If your mosaic is small these should be larger than the size of your final mosaic.


3.6 Running the Overlap Pipeline


Once you have the Overlap pipeline set up, click on the green arrow (play button) in the top left of the MOPEX window to start the pipeline. If you keep all intermediate data this will require 5.8GB of disk space for the ch1 COSMOS data.




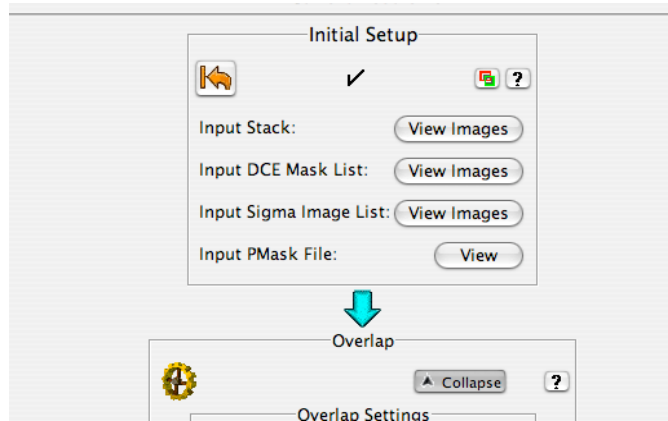
The pipeline will begin running modules in the order you specified. You can stop or pause the pipeline at any point by clicking on the stop or pause buttons.




You can also run the pipeline to a certain point by clicking the "Run Module" button  on the left of each module box.

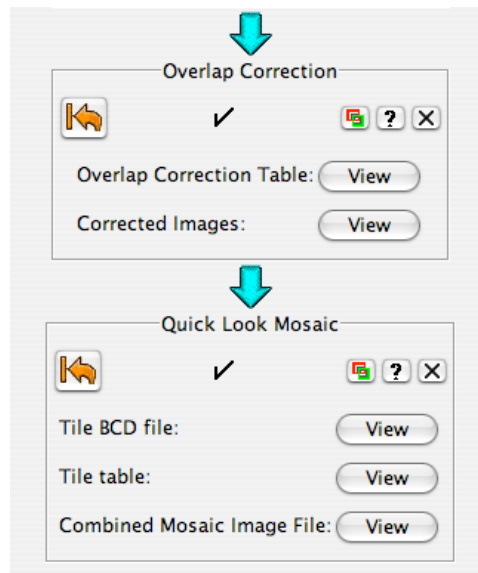
As the pipeline is running the current working component will have a turning gear  in the top left corner. The current working module will also be labelled with a gear and will show you its status at the bottom of the module box. When a module is finished a check mark will appear in the top center and a

"Go Back"  button will replace the "Run Module" button on the left. Clicking on this button will bring the pipeline back to this module. Note that running the overlap correction can take a long time.



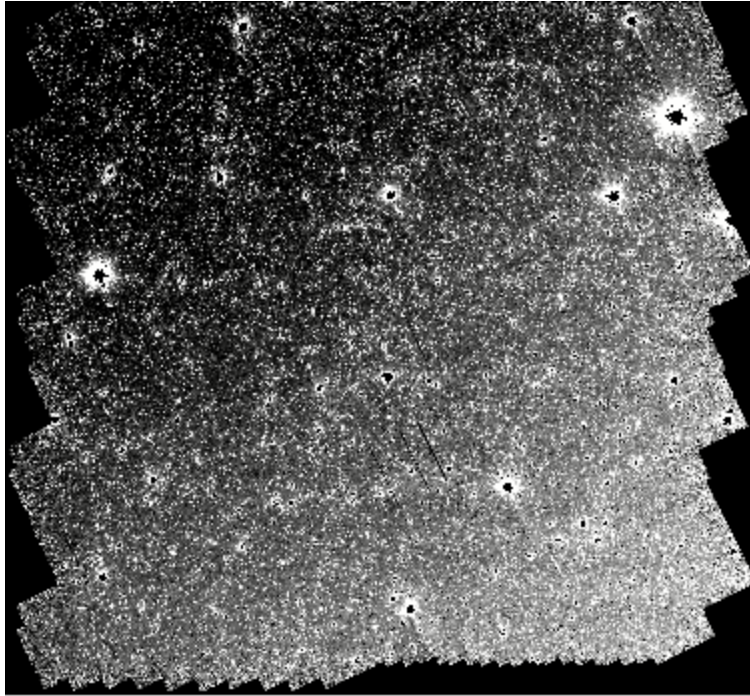
3.6.1 Checking Overlap Results

After overlap runs you will want to check the output for problematic frames and decide if you want to remove them or modify the pipeline. You can view the output from each module directly in MOPEX by clicking on the "View" buttons corresponding to the outputs in the module window once the module has finished running. You can also view the module log by clicking on the red and green "Log" button  on the right hand side of the module boxes.



A good place to start is by inspecting the mosaic frame. While you can view the image directly in MOPEX, the overhead associated with JAVA means it is usually faster and more flexible to open it in ds9 or another FITS viewer.

Below is a picture of the quick look mosaic for the ch1 COSMOS data.

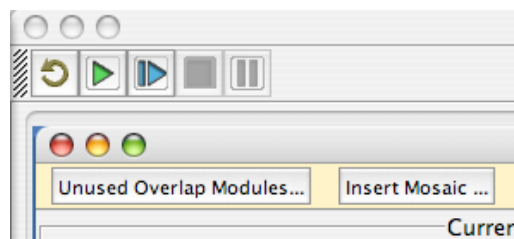


The gradient across the field is real and due to Zodiacal light.

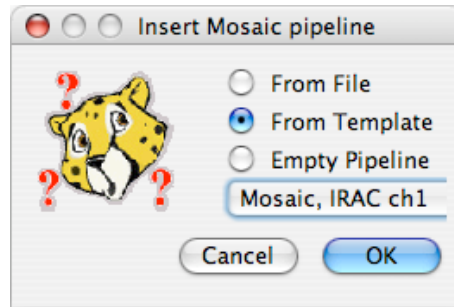
3.7 Channel 1 Mosaic

The Mosaic module finds and removes outliers from the individual images, re-samples them onto a common reference frame, and combines them into a final stacked image. There are three possible methods for outlier rejection: "Dual Outlier", "Mosaic Outlier", and "Mosaic Box Outlier". "Dual Outlier" is best suited for low coverage and is discussed elsewhere. "Mosaic Outlier" and "Mosaic Box Outlier" search for pixels which deviate from the median in time and space using multiple overlapping exposures.

Once you are happy with the output of the Overlap module, you will want to add a mosaic module to your pipeline. You can do this by clicking on "Insert Mosaic..." in the top left of the MOPEX window.

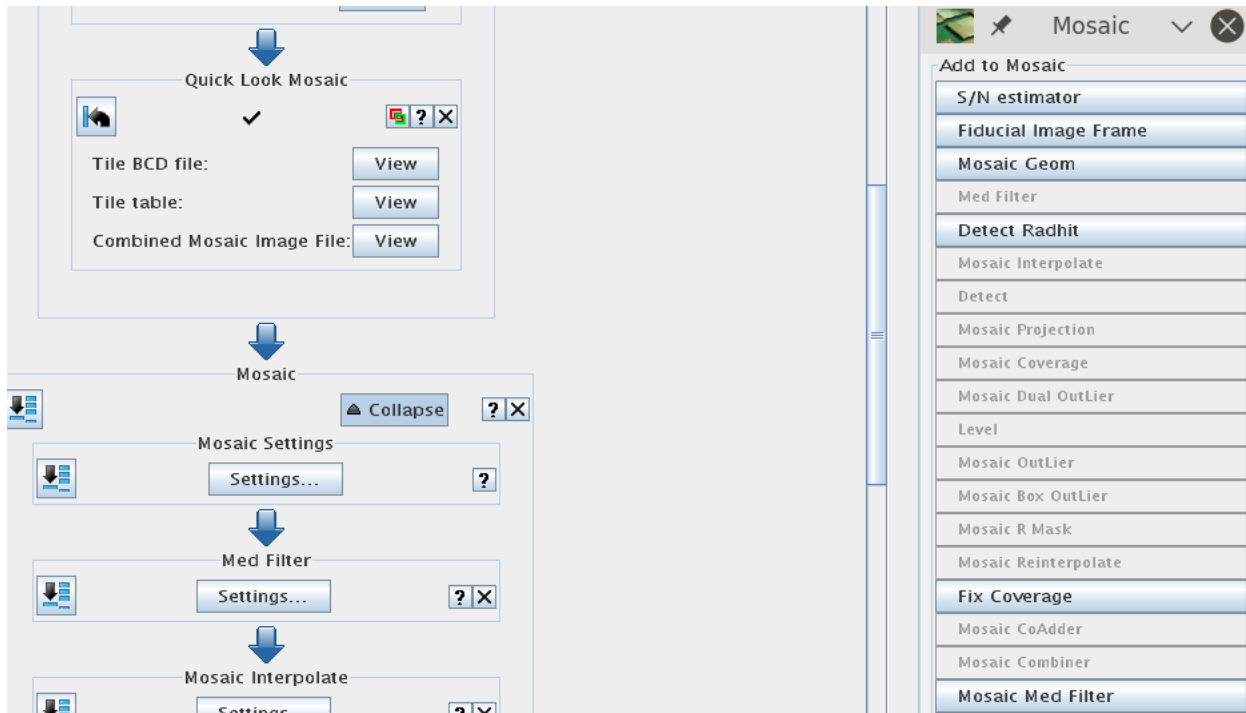


That will bring up the following window.

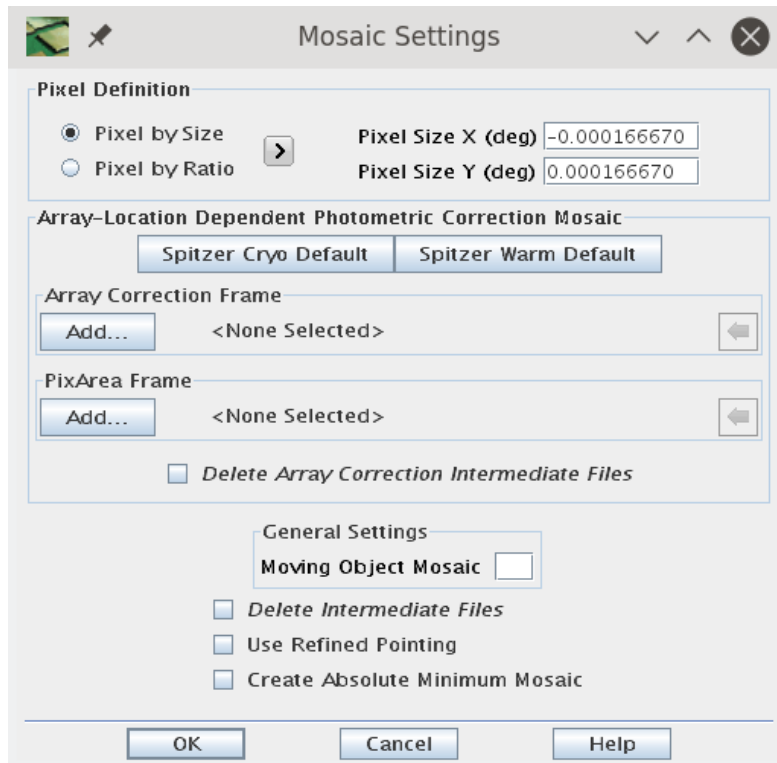


Check the "From Template" option and then "Mosaic, IRAC ch1" from the drop down menu.

The default mosaic module will be added to your pipeline after the end of the overlap module and a menu of possible mosaic modules will appear.



3.7.1 Mosaic Settings



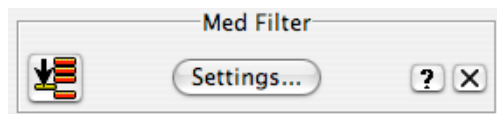
The "Mosaic Settings" dialog box sets several global parameters related to image combination and temporary files.

If you are short on disk space you may want to check the "Delete Intermediate Files" box, which will remove the intermediate data products when they are no longer needed.

Check "Use Refined Pointing" to use the pointing updates in the CBCD files. This is generally not required for data taken after Cycle-2. The pointing refinement involves identifying point sources in the IRAC frames and correlating them with 2MASS source positions. The pointing refinement typically improves the positional error to less than 0.3 arcseconds and removes any systematic offsets. For more information, see Section 5.3.1 of the IRAC Instrument Handbook.

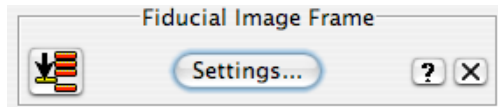
The other boxes correspond to data products not covered by this recipe.

3.7.2 *Med Filter*

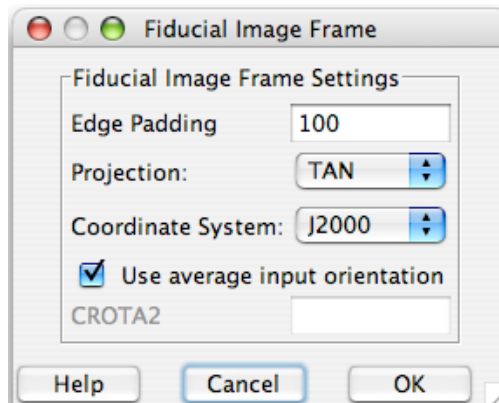


This module filters and removes the background as part of the "Dual Outlier" bad pixel rejection, which is optimal for very low coverage and is not used in this recipe. Remove this module from the pipeline by clicking on the "X" to the right of the box.

3.7.3 Fiducial Image Frame



The "Fiducial Image Frame" sets the parameters for the final mosaic. To configure it, click on "Settings..." in the "Fiducial Image Frame" box. The following dialog box will appear:



Edge padding specifies the number of blank pixels around the edge of the data. A blank area of this width will be placed on all sides of your final mosaic. The default value of 100 is fine for most cases.

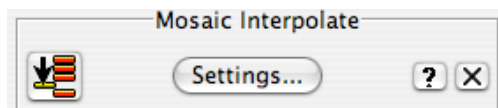
Projection specifies the WCS projection. The default "TAN" is used for most astronomical FITS images.

Coordinate System specifies the celestial system for the WCS. Options are J2000, Galactic or Ecliptic. For extragalactic studies, J2000 is the appropriate setting.

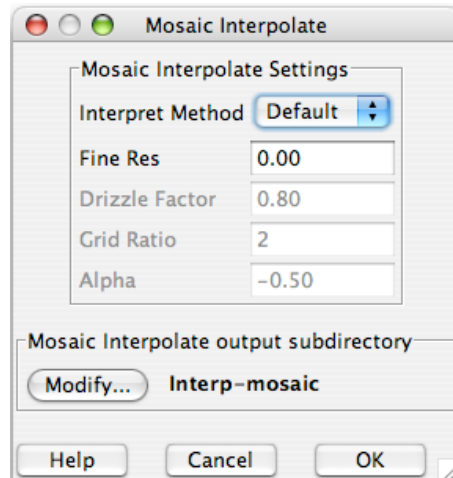
The "Use average input orientation" box specifies how the WCS rotation is specified. Leaving this box checked will minimize the size of the final fits image by making the final mosaic as rectangular as possible. However, it is often desirable to specify a fixed rotation to match other data. For COSMOS, uncheck this box and set CROTA2 to 0.00.

When you are finished, click "OK".

3.7.4 Mosaic Interpolation



The "Mosaic Interpolate" module does the first pass interpolation of the data onto the FIF. To configure this module, click on "Settings..." in the "Mosaic Interpolate" box. The following dialog box will appear:



If you have more than a few exposures of dithered data you may wish to use the Drizzle interpolation. This method attempts to recover full sampling of the PSF while minimizing the amount of correlated noise. Drizzle modifies the noise properties of the image in a non-linear manner. For $\text{pixfrac} > 0.01$ the noise will still be correlated, but at a lower level than a linear interpolation. If you are concerned about the noise properties of your images, you should use the default interpolation, which is well characterized, and correct for the noise correlation using the mosaic noise maps.

Drizzle operates by shrinking each pixel by the Drizzle factor, then projecting it onto the final pixel grid. The optimal choice of Drizzle factor depends on the original and final pixel size along with the amount of coverage. The smaller the Drizzle factor the better sampled your final PSF will be. However, if the factor is too small and there are not a sufficient number of exposures, you will not fill the final mosaic. In addition, there is no point in shrinking the pixel to a value smaller than the output pixel scale.

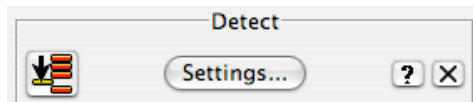
A factor of 0.65 is optimal for the COSMOS IRAC data with an output pixel size of 0.6".

Note that when using the Drizzle interpolation, several of the modules are run twice. In previous versions of MOPEX (releases earlier than 16.3.7), this had to be done manually, but the process is now included in the pipeline automatically. Firstly a linear interpolation is carried out so that MOPEX can run the outlier rejection scheme. Once the outlier rejection masks (RMasks) have been created, MOPEX returns to the Interpolate module and re-runs the interpolation with the Drizzle algorithm, masking out any pixels flagged in the RMasks.

When using Drizzle you will need to remove the "Mosaic Reinterpolate" module later in the workflow.

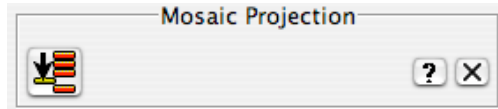
The "Mosaic Interpolate output subdirectory" specifies the location of temporary files.

3.7.5 Detect



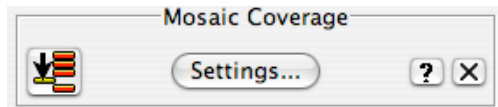
This module detects objects as part of the "Dual Outlier" bad pixel rejection scheme, which is optimal for very low coverage and is not used in this recipe. Remove this module from the pipeline by clicking on the "X" at the right of the box.

3.7.6 *Mosaic Projection*



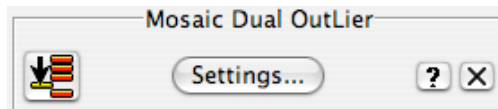
This module projects the detection map onto the final mosaic as part of the "Dual Outlier" bad pixel rejection scheme, which is optimal for very low coverage and is not used in this recipe. Remove this module from the pipeline by clicking on the "X" at the right of the box.

3.7.7 *Mosaic Coverage*



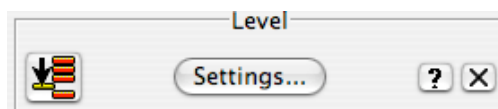
This module configures the image size used for intermediate data products. If you have lots of memory (more than 1GB) you can increase this from the default value and if you are running out of memory you can decrease it at the cost of speed.

3.7.8 *Mosaic Dual Outlier*



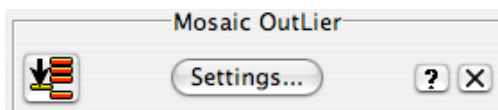
This module conducts the "Dual Outlier" bad pixel rejection scheme, which is optimal for very low coverage and is not used in this recipe. Remove this module from the pipeline by clicking on the "X" at the right of the box.

3.7.9 *Level*

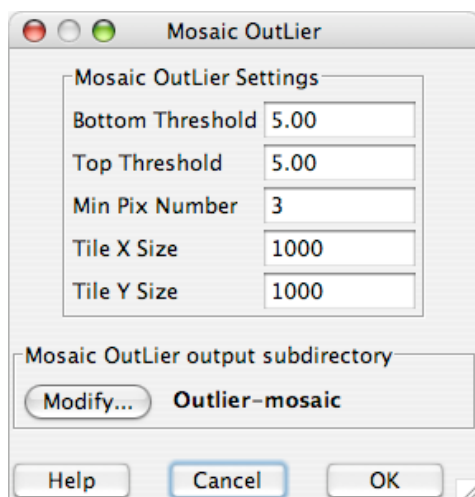


This module removes false outliers around real point sources as part of the "Dual Outlier" bad pixel rejection which is optimal for very low coverage and is not used in this recipe. Remove this module from the pipeline by clicking on the "X" at the right of the box.

3.7.10 *Mosaic Outlier*



This module uses multiple frames to reject bad pixels. It measures the median and sigma for each pixel in the final mosaic and flags stray pixels. This method works best when you have coverage greater than 10 so that the sigma and median are well determined.



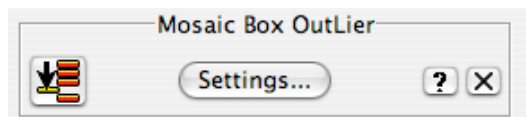
The "Bottom Threshold" and "Top Threshold" specify the number of sigmas where flagging occurs.

"Min Pix Number" specifies the minimum number of frames required to measure sigma directly.

The tile X and Y size specify the size of the working mosaic in memory.

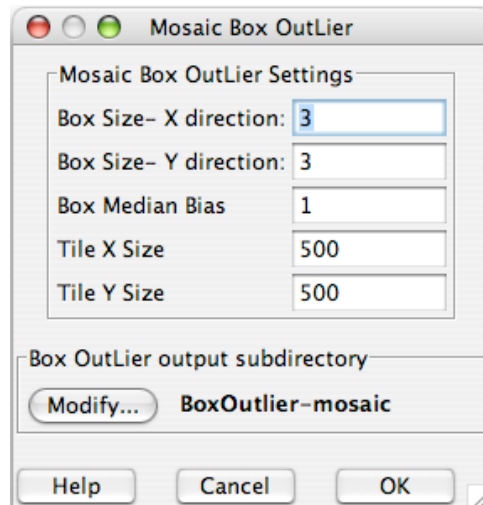
The default settings are optimal for a coverage of ~10, but if you have higher coverage you should consider increasing the "Min Pix Number" and decreasing sigma.

3.7.11 Mosaic Box Outlier



This module preforms outlier rejection using a method similar to "Mosaic Outlier", but considers a window around each pixel rather than just a single pixel. This is useful for moderate coverage where "Mosaic Outlier" may miss some real outliers.

You will need to check the "Use Box Outlier for Rmask" setting in the "Mosaic R Mask" module for the results to propagate through to the final mosaic.



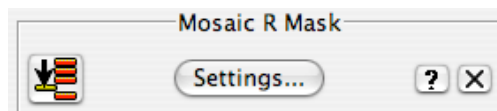
The "Box Size" settings specify the size of the window used to calculate the median in mosaic pixels.

The "Box Median Bias" setting allows you to bias the median used for rejection below the true median by a set number of pixels. This has the effect of rejecting any positive outliers from the median.

The "Tile X Size" and "Tile Y Size" settings control how the final mosaic is broken up during processing to save memory.

The default settings are fine for a final pixel scale of 0.6 arc seconds. If you have a very fine mosaic pixel scale you may want to increase the "Box Size" and "Box Median Bias" settings. You can increase the tile size settings if you have lots of memory.

3.7.12 *Mosaic R Mask*



This module combines the pixel masks and the various outlier rejection masks to generate a master bad pixel mask and a final bad pixel mask for each frame.



"RM Thresh" specifies the minimum fraction of an input pixel covered by a rejected pixel in the final projection for flagging to occur on the input image. Decreasing this value will improve outlier rejection, but also remove a larger number of good pixels. If you have a very fine final pixel scale you should decrease this value.

The "Bottom Threshold" and "Top Threshold" settings specify the upper and lower leaves in sigma for removing pixels flagged by "Mosaic Outlier". The default values are optimal for a coverage of ~10.

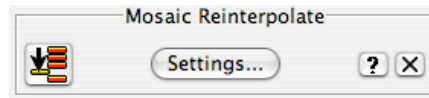
The "Min Coverage", "Max Coverage", "Refine Outlier", and "Refine Outlier Threshold" settings only have an effect if you used "Dual Outlier". Setting "Refine Outlier" to anything but zero will cause an error message if the various "Dual Outlier" modules were not run.

The "Box Bottom Threshold" and "Box Top Threshold" settings specify the upper and lower leaves in sigma for removing pixels flagged by "Mosaic Box Outlier". The default values are optimal for a coverage of ~10.

"Box Min Coverage" specifies the minimum number of overlapping frames for pixels flagged by "Mosaic Box Outlier" to be removed.

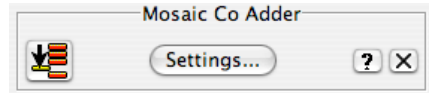
The three check boxes at the bottom of the window specify which outlier rejection methods will be considered by "Mosaic Rmask". Check "Use Outlier for Rmask" and "Use Box Outlier for Rmask", but uncheck "Use Dual Outlier for Rmask".

3.7.13 *Mosaic Reinterpolate*

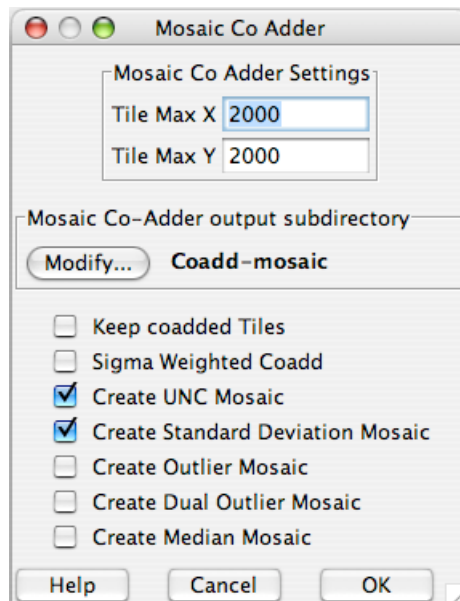


This module re-interpolates the raw data removing pixels flagged by "Mosaic Rmask". This module is not needed if "Drizzle" was selected in the "Mosaic Interpolate" module. Remove this module from the pipeline by clicking on the "X" at the right of the box.

3.7.14 Mosaic Co Adder



This module combines the masked, re-interpolated images into a final mosaic image. You can control how the co-addition is done and the size of the output images.



The "Tile Max X" and "Tile Max Y" keywords specify how the field is broken up to avoid generating very large files and using too much memory.

The "Mosaic Co-Adder output subdirectory" specifies where the files will be saved.

Checking "Keep co-added Tiles" will keep the tiled images which can be useful if your final image is very large.

Checking "Sigma Weighted Coadd" will generate a weighted average image using the uncertainty maps as the weight. If this is not checked an average image is generated.

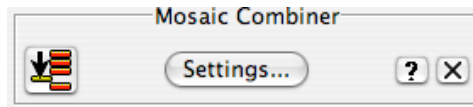
Checking "Create UNC Mosaic" will create a mosaic of the uncertainty maps.

Checking "Create Standard Deviation Mosaic" will create an image with the standard deviation of the input pixels as the value at each position.

Checking "Create Outlier Mosaic" or "Create Dual Outlier Mosaic" will cause the relevant bad pixel maps will be mosaicked together. This is useful for determining if real objects were rejected by the outlier detection methods.

Checking "Create Median Mosaic" will generate a median image. This does not conserve flux.

3.7.15 Mosaic Combiner



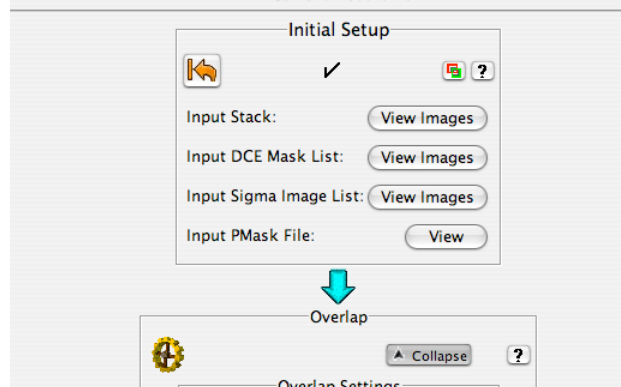
This module combines the mosaic tiles into a single final image. The only setting is the directory where you would like the images saved.

3.8 Other Channels

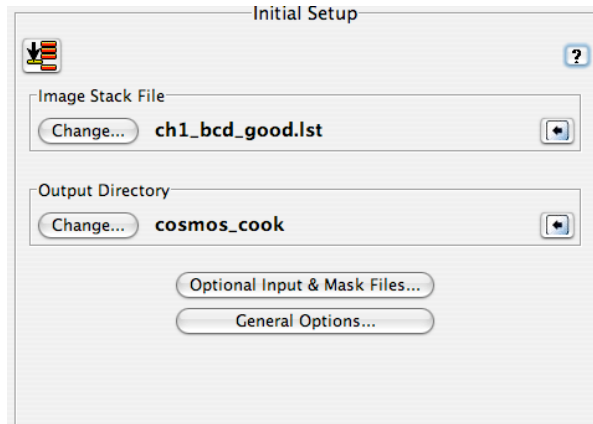
In most cases the settings used for channel 1 should be identical for the other three channels. The only changes needed are the names of the input files and removing the "Fiducial Image Frame" modules if you want the images to have the same coordinate system. However, you should verify that the object size and background dependent settings in the Overlap "Med Filter" and "Detect" modules are generating good results.

Before starting the other channels save the channel 1 namelist by clicking on "Save" or "Save As..." in the drop down file menu of MOPEX. Then revert to the beginning of the workflow by clicking on the "Go

Back"  button in the top left of the "Initial Setup" box at the beginning of the workflow.

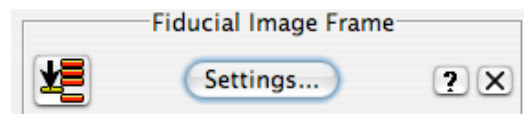


This will restore the pipeline to the beginning of the workflow. At this point its a good idea to save the setup with a different name to avoid accidentally overwriting the channel 1 setup. This can be done by clicking on "Save As..." in the drop down file menu of MOPEX.



To change the input files, click on "Change..." under "Image Stack File" in the "Initial Setup" box. Select the list of images for the channel you wish to process. Leave "Output Directory" with the same setting as channel 1. The corresponding "Sigma List File" and "DCE Status Mask List" files must also be changed under "Optional Input & Mask Files...". Finally, the "Pmask FITS File" should be changed from the channel 1 bad pixel mask to the relevant channel bad pixel mask.

If you want the pixel positions in the final images to correspond to the same sky positions in all four channels you must remove the "Fiducial Image Frame" module from both the Overlap and Mosaic pipelines. This can be done by clicking on the "X" at the top right of the FIF window. If you change the output directory, you will need to specify the FIF file under "Optional Input & Mask Files..." as the one generated during the channel 1 processing.



Once you have made these changes you can start the pipeline and the next channel will be processed.

3.9 Different Exposure Levels

The above recipe was designed to reduce medium deep (~15 minute integration with more than 10 exposures), blank field IRAC survey data. You will need to change some parameters if your data are of substantially different depth.

Shallower Data:

Change interpolation to default

Add dual outlier mosaic to rejection

Update mosaic rmask to add dual outlier mosaic rejection

For Galactic case with diffuse emission, change settings in mosaic rmask to refine thresholds based on the various outlier rejections

HDR mode:

Same as shallow.

In addition, modify mosaic-rmask pixel fraction rejection to not remove pixels unless a large fraction is flagged. This prevents the low exposure data from being over-flagged.

Deeper Data:

Decrease thresholds for box and temporal outlier.

Decrease pixfrac in drizzle.

Decrease fraction of pixel to be flagged in mosaic-rmask.

Recipe 4. MOPEX: Mosaic IRAC Galactic FLS Data

Here we use one of several AORs which comprise the Galactic First Look Survey (GFLS; <http://irsa.ipac.caltech.edu/data/SPITZER/docs/spitzermission/observingprograms/firstlooksurvey/>). This survey was taken in IRAC's high dynamic range (HDR) mode (see the IRAC Instrument Handbook for more information on this mode). This means the even (odd) numbered CBCD files will contain images taken with, in this case, 0.4 (10.4) second exposure times, or, equivalently, 0.6 (12) second frame times. We explain how to make a channel 3 (5.8 micron) mosaic from the longer frame times. In particular, we focus on the delta offset correction that must commonly be made for channel 3 data. We also describe which outlier rejection and image interpolation schemes are appropriate for low coverage data (in this case, 4 images per point on the sky).

You may also be interested in reading the IRAC COSMOS recipe (Recipe 3), which demonstrates how to reduce IRAC channel 1 data with medium coverage. The COSMOS recipe goes into greater detail about the different parameters that can be set, so will be helpful even to those users with low coverage data.

4.1 Requirements

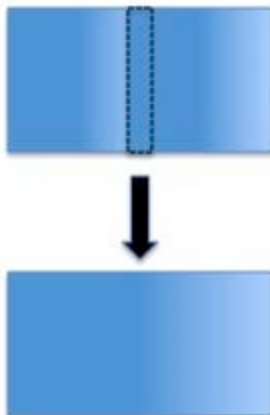
You must have the Spitzer software package MOPEX installed in order to follow along with this recipe (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/mopex/>). This demonstration was originally tested using MOPEX 18.2. Use the Spitzer archive to download the BCD and post-BCD data for AOR 4960000 for this example. See Recipe 1 for instructions on downloading the data.

4.2 Correct BCDs for residual offset

What is the residual offset? Even in the absence of any photons, IRAC images will exhibit non-zero (usually positive) flux. The amount of flux varies across a given image. It also depends on the frame time and the Fowler number, as well as the history of readouts and array over the previous several hours. The effect is largest in the first frame of an observation, which is why the first frame is always taken in HDR mode, wherein a very short exposure (which you should definitely discard) is followed by a long one (which you may be able to use). However, the residual offset exists at a lower level for all frames, especially in IRAC channel 3, but it is substantial also in channels 4 and 1. Thus, you may need to correct for this residual offset. For more information, see the IRAC Instrument Handbook.

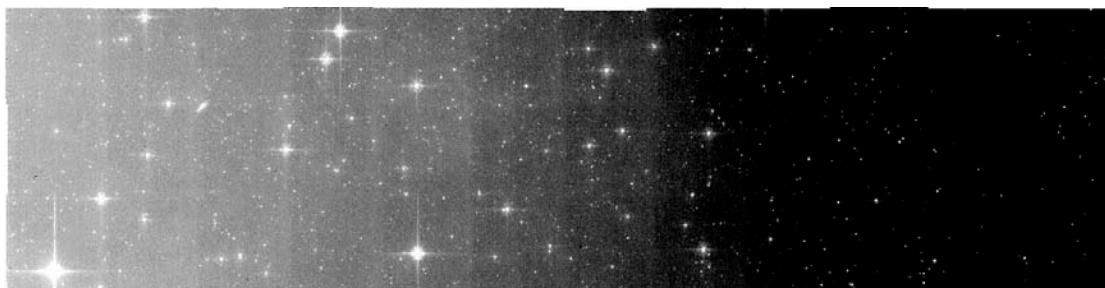
How can you tell if your data require a residual offset correction? In general, it is not a good idea to make a correction that you do not really need, as this may actually make your data worse by introducing noise. Therefore, it is important to determine whether your particular data set requires a residual offset correction. An easy way to make this assessment is to first try reducing your data without the correction, and look for problems.

A significant (uncorrected) residual offset in your data can result in a gradient across your overlap-corrected mosaic. This is because the residual offset tends to vary systematically across an individual CBCD. Therefore, when the overlap module tries to figure out a single offset for each CBCD in order to match overlapping regions, a systematic gradient across the whole mosaic results. Here is a cartoon illustrating the effect:



The top image above the arrow represents two overlapping CBCDs, each with its own residual offset gradient. The dotted rectangle highlights the overlapping region. The image below the arrow represents the mosaic made after applying an overlap correction. The overlap correction required a relative decrease in the flux in the right-hand CBCD in order to make the overlapping region the same in both CBCDs.

What would our example data set look like if we did not correct for residual offsets? We can take the ch 3 pipeline mosaic (PBCD file SPITZER_I3_4960000_0000_8_E8349728_maic.fits) or run our original downloaded CBCDs through the overlap pipeline, and sure enough, we see a strong gradient, justifying the need for a residual offset correction:



If you would like to reproduce the image above, follow the instructions in Section 3.6, but change the output directory to `overlap_output` (so you do not overwrite it later) and use the following files for your Image Stack File, your Sigma List File, and your DCE Status Mask List, respectively (do not forget to edit them with the path to the data on your computer):

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/ch3_10.4_cbcd.list
http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/ch3_10.4_cbunc.list
http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/ch3_10.4_bimsk.list

How do I correct for residual offsets? To get rid of the artificial gradient seen above, we recommend creating a delta offset frame, subtracting it from each BCD, and using the results in all subsequent processing (i.e. MOPEX overlap and mosaic pipelines).

As described in the following list, there are several ways to create a delta offset frame. Regardless of which way you choose, for this example you should not use any of the even-numbered frames, which have short exposure times. Remember that we are making a mosaic of the long exposure times (header keyword HDRFRAME = long), so you must use a subset of the odd-numbered frames. Also keep in mind that MOPEX does not create residual offset frames. You must use the image manipulation software of your choice, e.g. IRAF, IDL.

1. The simplest possible delta offset frame is a median of all of the input CBCDs. This requires all CBCDs to have nearly the same overall sky level. Also, this method will subtract the background as well as the residual offset.
2. Another slightly more complicated option is to first subtract the median of each frame from itself. Then compute the delta residual offset as the stack median of the resulting frames. Finally, subtract this stack median from the original downloaded CBCDs. Using this method removes the slope associated with both the residual offset and the background. However, this method will not remove any constant offset due to the residual offset or the background.
3. As mentioned above, the residual offset is a function of the time between subsequent frames. For instance, if you take two images in a row, the residual offset in the second frame will be larger if you are doing a repeat rather than dithering between frames. If you have enough frames, you can try making multiple delta offset frames (using either of the above procedures), one for each repeat. A simple IDL code (http://irsa.ipac.caltech.edu/data/SPITZER/docs/files/spitzer/repeat_delta_dark.pro) for doing this is available. An example of how to use the code on IRAC data is provided in the header of the IDL procedure.

In this example, we use the first method, as our frames have about the same background level; we are not interested in the background intensity; and we do not have any repeats. We encourage you to try the different methods to see the effect on your final mosaic. We gave the delta residual offset subtracted CBCDs the same names as the original CBCDs with an added `do_` prefix, for "delta offset".

4.3 Set up files in preparation to run MOPEX

Before we can run MOPEX to compute the overlap correction and remosaic the data, we need to do a little preparation.

- In the directory `/where/you/unpacked_data/`, make subdirectories `cal/` and `cdf/`. The `cal` directory will hold all of our calibration files, and the `cdf` directory will hold the namelists which save our MOPEX parameters.

- Make a list of the CBCD, uncertainty, and bad pixel mask files in the directory */where/you/unpacked_data/*. Again, the lists should include only the odd-numbered frames, which have long exposure times (header keyword HDRFRAME = long). In addition, each line should contain the full path to the data. Finally, do not forget that your CBCD list should consist of your **residual offset corrected CBCDs** (with prefix *do_* in this example), not the pipeline-produced CBCDs that you originally downloaded. For your convenience, we provide the necessary files here:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/ch3_10.4_cbcd_offsetcorr.list

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/ch3_10.4_cbunc.list

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/ch3_10.4_bimsk.list

Remember to edit these files to reflect the path to the data on your computer.

- Download the pmask calibration files (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/>) to your *cal/* directory. These files will be used to mask out bad pixels in the array, so that they are not included in your final mosaic. These pmaps are updated intermittently, on timescales of weeks to months, depending on how rapidly the bad pixels are seen to change.

4.4 Compute the overlap correction

The Overlap Pipeline removes any remaining offsets in CBCD background by calculating an additive correction for each cbcd image to bring them to a common background level. Follow these steps to run the overlap pipeline on the residual offset corrected CBCDs.

1. Start up the MOPEX GUI.
2. Open up a MOPEX Overlap Flow. Click on "File"-->"New Overlap Pipeline..."-->"Overlap, IRAC ch3"--"OK". You will see the previously empty MOPEX window fill up with the Overlap flow.
3. Set parameters in MOPEX Overlap Flow.

Below is the minimum set of parameters that should be set in running overlap. A full description of each of the parameters is provided in the MOPEX manual and is also discussed in the IRAC COSMOS Recipe (Recipe 3).

 - a. Image Stack File should be the *ch3_10.4_cbcd_offsetcorr.list* file you downloaded and edited previously.

- b. Output Directory is your choice, but here we choose
/where/you/unpacked_data/overlap_offsetcorr_output
 - c. Optional Input & Mask Files-->Sigma List File should be the *ch3_10.4_cbunc.list* file you downloaded and edited previously.
 - d. Optional Input & Mask Files-->DCE Status Mask List should be the *ch3_10.4_bimsk.list* file you downloaded and edited previously.
 - e. Pmask FITS file - To decide which pmask to use, you must look at the date your CBCDs were observed. Looking at the DATE_OBS keyword in the fits header of one of the CBCD files, we find that these data were taken on December 6, 2003. Therefore, we should use the file */where/you/unpacked_data/cal/mar04/mar04_ch3_bcd_pmask.fits*.
 - f. Click on the Quick Look Mosaic in MOPEX to add that module
4. Save your Overlap Flow parameters. In the main MOPEX menu, choose "File"-->"Save 'untitled' As...". Then choose the directory */where/you/unpacked_data/cdf/overlap_offsetcorr.nl*
 5. Run the Overlap Flow. Simply click on the green arrow at the top left of the MOPEX window. The GUI will keep you apprised of its progress.

The resulting mosaic produced by overlap can be found in
/where/you/unpacked_data/overlap_offsetcorr_output/mosaic.fits.

It should look something like this:



Notice that there is no longer a systematic gradient across the field, as there was when we ran overlap on CBCDs that had not been corrected for the residual offset. The mosaic still has imperfections, but the overlap mosaic is not meant to be science-grade. It is merely meant to compute the offsets between CBCDs. We can see from this image that it has been successful. In the next step, we will make a science-grade mosaic.

4.5 Produce a Science Mosaic

Now that we have corrected the CBCDs for both the residual offset and mismatched backgrounds, the next step is to use the corrected CBCDs to produce a new mosaic.

1. Start up the MOPEX GUI.
2. Open up a Mosaic Pipeline. From the main menus, select File-->New Mosaic Pipeline-->Mosaic, IRAC ch3-->OK.
3. Set Mosaic Parameters.
 - a. For the image stack file, you should select the stack of overlap-corrected images. This will appear in the overlap output directory. In our case, this
/where/you/unpacked_data/overlap_offsetcorr_output/Overlap_corr/overlapCorrectionList.txt
 - b. For the output directory, choose */where/you/unpacked_data/mosaic_default_output*.
 - c. In "Optional Input & Mask Files"-->"Sigma List File", choose
/where/you/unpacked_data/ch3_10.4_cbunc.list.
 - d. In "Optional Input & Mask Files"-->"DCE Status Mask List", choose
/where/you/unpacked_data/ch3_10.4_bimsk.list.
 - e. In "Optional Input & Mask Files"-->"Pmask FITS File", choose
/where/you/unpacked_data/cal/mar04/mar04_ch3_bcd_pmask.fits.
 - f. The most important parameters in the rest of the flow are those that control outlier rejection and image interpolation.

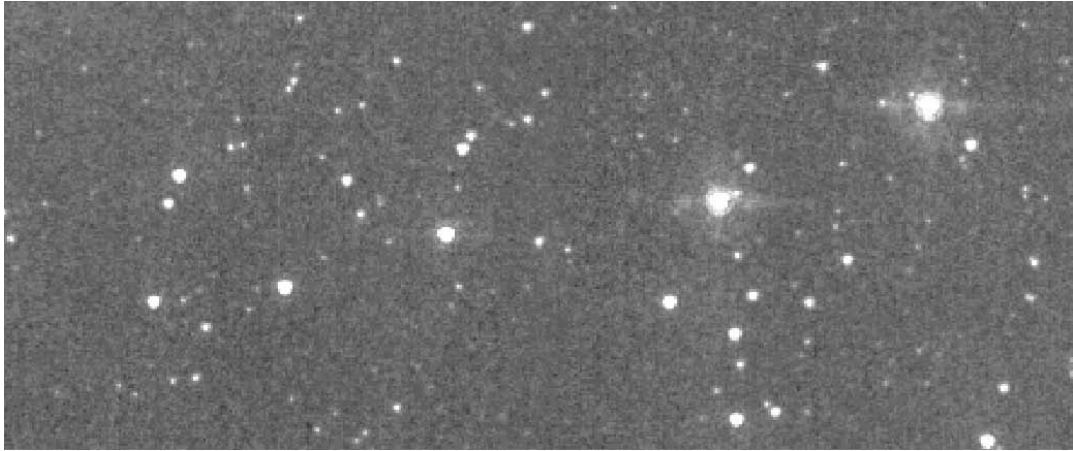
As described in the "Basic Concepts: Outlier Detection" section of the online MOPEX manual, MOPEX has four algorithms for identifying bad pixels: single frame, multiframe temporal, dual outlier, and box outlier. The default IRAC namelist uses dual outlier. For low coverage data like ours (the AOR we have chosen has a coverage of 4), we recommend using either dual outlier or box outlier. Since both methods tend to produce similar results, we will remove the "Mosaic Outlier" module and in the Mosaic R Mask module will uncheck the "Use Outlier for Rmask" option.

As described in the MOPEX manual, four interpolation schemes are implemented: Default, Drizzle, Grid and Cubic. General guidelines for choosing the interpolation scheme appropriate for your data set are provided in the [Helpdesk](http://sohelp2.ipac.caltech.edu/support/index.php?Knowledgebase/Article/View/68) knowledgebase: <http://sohelp2.ipac.caltech.edu/support/index.php?Knowledgebase/Article/View/68>. A rule of thumb is that if you have a coverage of 10 or less (as we do), you will probably want to use the default interpolation scheme.

If you have higher coverage data than the low coverage data we are analyzing here, you may wish to use a different outlier rejection scheme and a different interpolation method. For a medium coverage example, see the IRAC COSMOS recipe (Recipe 3).

4. Run the Mosaic Pipeline. Simply click on the green arrow.
5. Examine the results. After running the mosaic pipeline, you can find the output in
/where/you/unpacked_data/mosaic_default_output/Combine-mosaic/mosaic.fits

6. The image looks fairly good. Here is a zoomed-in section in the middle of the image, demonstrating that the outlier rejection worked properly:



Recipe 5. MOPEX: IRAC Mosaic of the galaxy NGC1097

This example demonstrates how to use MOPEX to create mosaics of a large field of view around the SINGS Legacy Science Program's galaxy NGC 1097 in IRAC channels 1 through 4. We pay special attention to choosing the appropriate parameters in the MOPEX namelist files, and discuss the effect of various choices of several of these parameters. The final products include the mosaic image, the mosaic uncertainty image, and the mosaic coverage map. Note that this example uses the command line way of running the mosaicer instead of the GUI that was used in previous sections.

5.1 Requirements

You must have MOPEX installed in order to follow along with this recipe (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/tools/mopex>).

You will need to use these namelists:

- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch1overlap.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch1mos.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch2overlap.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch2mos.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch3overlap.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch3mos.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch4overlap.nl>
- <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/cookbook/files/ngc1097ch4mos.nl>

5.2 Downloading the example data set

Download the BCD and calibration data for channels 1 through 4 from AORs 5515776 and 5516032. Put all the data into directories of your choice - you can choose to rearrange the directories that the Spitzer archive sets up for you, for example, by combining all of the BCD data from two AORs into one directory.

You need the input CBCD files, listed in a file (here called *image_stack_all.txt*, one file per line). Only files with odd expids, (e.g., only the 30 sec frames in the HDR observations), should be included. Here is

a nifty unix trick to just get the odd- or even-numbered files in a file, one file per line (assuming you are currently in the directory with all the CBCDs):

```
unix% ls *cbcd.fits | awk 'NR % 2 == 0 { print }' > oddlist.txt
unix% ls *cbcd.fits | awk 'NR % 2 == 1 { print }' > evenlist.txt
```

You also need a similar list of the corresponding uncertainty files (**cbunc.fits* files) and imask files (**bimsk.fits* files). You can also simply type “`ls *cbcd.fits > filelist.txt`” and edit out the short exposures, (e.g., keep *SPITZER_II_5515776_0001_0000_4_cbcd.fits* but delete *SPITZER_II_5515776_0000_0000_4_cbcd.fits*, keep *SPITZER_II_5515776_0003_0000_4_cbcd.fits* but delete *SPITZER_II_5515776_0002_0000_4_cbcd.fits* etc.). Do the same for the uncertainty and imask files.

Do not forget to also download the IRAC Pmask files (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/>).

There are several artifact correction software packages at

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyisistools/tools/contributed/irac/>

that you can use to get rid of any remaining artifacts in the CBCD images before making the mosaics.

5.3 Perform Overlap Correction

It is recommended that you run the frames through an overlap correction module before proceeding with mosaicking.

```
unix% overlap.pl -n ngc1097ch1overlap.nl
unix% overlap.pl -n ngc1097ch2overlap.nl
unix% overlap.pl -n ngc1097ch3overlap.nl
unix% overlap.pl -n ngc1097ch4overlap.nl
```

The links to the namelists to be used (remember to change the paths to input and output files to correspond to your directories) are given above in the "Requirements" section. Make sure that you have specified the input CBCDs in the file pointed to by the `IMAGE_STACK_FILE_NAME` parameter in the namelist file (including their path on your disk), and similarly the `SIGMALIST_FILE_NAME` and `DCE_STATUS_MASK_LIST` should point to files that give the full path and filenames to your uncertainty and imask images, respectively (one path/filename per line). For *overlap.pl*, you also need to give the `OVERLAP_CORR_DIR` name in the namelist file. This is the name of the directory, under your specified `OUTPUT_DIR`, where the final, overlap-corrected CBCDs will be written to. Pay attention to specifying the parameters for the overlap correction:

```

&COMPUTEOVERLAPCORRIN
  TOP_THRESHOLD = 3,
  BOTTOM_THRESHOLD = 3,
  MIN_IMG_NUM = 4,
&END

```

These parameters are used in the outlier rejection to set the upper and lower limits (multiples of sigmas), and the minimum number of images needed to perform the overlap correction. Also remember to set `COMPUTE_OVERLAP_CORRECTION = 1` and `APPLY_OVERLAP_CORRECTION = 1` in the namelist file. Finally there is a parameter `MASK_BRIGHT`. You can set this to 1 to use masks created by the “detect” module. In this example, the central parts of the galaxy are very bright, and adjusting the parameters to detect less bright objects only did not help to avoid the masking of the brightest galaxy parts, and therefore the `MASK_BRIGHT` parameter is set to 0. You can experiment by running the overlap correction module with `MASK_BRIGHT` set to 1. In addition, the first frame of an AOR often has a bias offset which is very different from the rest of the frames in the AOR. In this example, we leave the first frames (`EXPID=1`) in channel 3 out from both AORs in generating our overlap-corrected frames and mosaics. Most of the rest of the parameters specified in the namelist of the overlap correction module are discussed below in the context of generating the mosaics. Remember that your `IMAGE_STACK_FILE_NAME` when making the mosaics below must point to a file that specifies the overlap-corrected filenames (and their path on your disk; filenames starting with “*correct_SPITZER*”), not the original CBCDs. Also note that the option `MOSAIC_CORRECTED_IMAGES` will only give a quick and dirty mosaic to see whether the overlap correction worked okay. A thorough mosaicking with the *mosaic.pl* script, as described below, is necessary to produce a high quality mosaic.

5.4 Make the mosaic

Generate the fiducial frame.

This creates the file *FIF.tbl* that gives the header keywords that specify the orientation, size, etc. of the final mosaic (`CRVAL,CRPIX,CDELTA,NAXIS`). It also creates the file *header_list.tbl* that lists the projected `CRVAL, CRPIX` etc. values for each input BCD image. Specify the items in the left-hand column below (explanations are on the right) in the namelist file:

```
run_fiducial_image_frame = 1
```

```
Edge_pagged = 100
```

Add 100 arseconds of NaN-valued pixels on each edge.

```
CROTA2 = A
```

Orient the final mosaic. Use A for average CROTA values of the input CBCDs, 0 (“zero”) for north up orientation. If not specified, an optimal orientation is used that minimizes the area covered by the mosaic.

```
MOSAIC_PIXEL_SIZE_X =
```

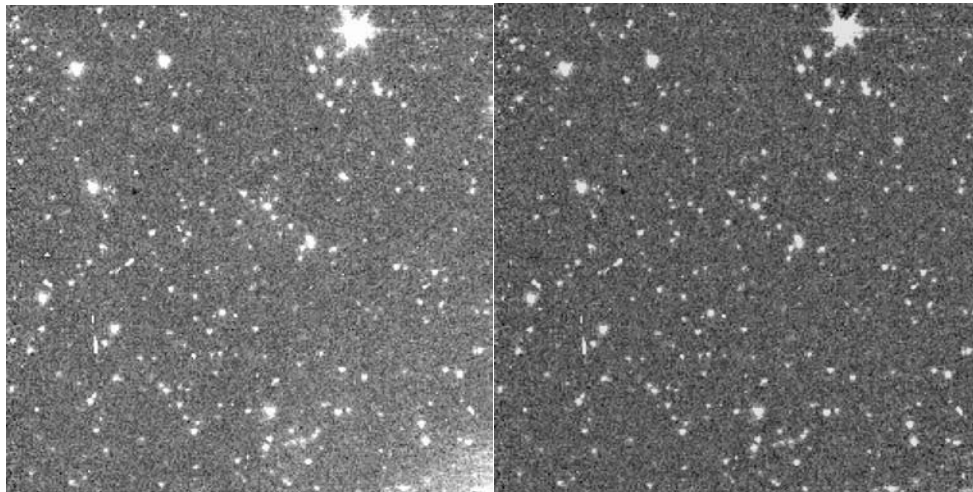
```
-0.00016667  
MOSAIC_PIXEL_SIZE_Y =  
-0.00016667          Set the pixel size of the mosaic.
```

Subtract the background in the input images.

Background subtraction is performed by subtracting median-filtered values from pixels. Background subtraction needs to be run only if single or dual outlier rejection is to be run. You need to specify the following in the namelist:

```
run_medfilter = 1  
  
&MEDFILTER  
Window_X = 21,  
Window_Y = 21,  
N_Outliers_Per_Window = 100,  
Use_Sbkg_for_Med = 0,  
&END
```

Trying a larger Window size (and larger outlier number) does not produce as good results. Trying a smaller Window size did not remove the background.



Ch 1 BCD (left) and Ch 1 image after background subtraction (right).

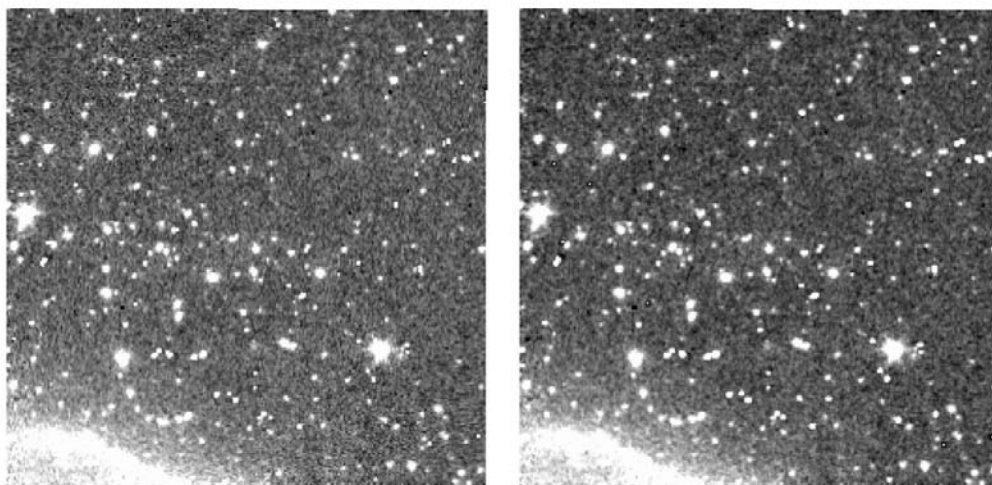
Interpolate the output pixels based on input pixel values.

The following parameters need to be set in the namelist:

```
run_mosaic_interp = 1  
  
&MOSAICINTIN  
INTERP_METHOD = 1,  
FINERES = 0,
```

&END

It is also possible to do drizzling (INTERP_METHOD = 2) or bicubic interpolation (INTERP_METHOD = 4). For drizzle, specify drizzle factor (DRIZ_FAC < 1). In channels 3 and 4 a better mosaic results if the FINERES parameter is not used, or if you use drizzle or bicubic interpolation.



Interpolation using the default option (left) or drizzling (right).

Outlier rejection

For dual outlier (spatial and temporal) rejection

```
run_medfilter = 1
run_detect_outlier = 1
run_mosaic_proj = 1
run_mosaic_dual_outlier = 1
run_level = 1
USE_DUAL_OUTLIER_FOR_RMASK = 1
```

For run_detect_outlier, the parameter list is DETECT:

```
&DETECT
Detection_Max_Area = 100,
Detection_Min_Area = 0,
Detection_Threshold = 4,
&END
```

Try playing with the threshold parameter. Increasing it causes a lot of cosmic rays to not be detected. Lowering it causes too many detections. Leave it at 4.

For dual outlier detection, the parameters are

```
&MOSAICDUALOUTLIERIN
MAX_OUTL_IMAGE = 2,
MAX_OUTL_FRAC = 0.5,
TILE_XSIZ = 500,
TILE_YSIZ = 500,
&END
```

Play with MAX_OUTL_FRAC. Increase it to 0.95. You will see a lot of non-radhit detections. If you lower it to 0.2, very few detections are seen. The best value seems to be 0.5. Tile size parameters can be played with if your computer has problems with insufficient memory.

For run_level the parameter is THRESHOLD_RATIO. Use 0.5 to have all the pixels within a cluster to have the same sign (see mosaicker user's guide for more info).

For box outlier (spatial and temporal) rejection add the following lines to your namelist file:

```
run_mosaic_box_outlier = 1
USE_BOX_OUTLIER_FOR_RMASK = 1
```

The parameters to be added to the namelist file are

```
&MOSAICBOXOUTLIERIN
BOX_X = 3,
BOX_Y = 3,
BOX_MEDIAN_BIAS = 1,
TILE_XSIZ = 500,
TILE_YSIZ = 500,
&END
```

Trying a box size of 5 to enlarge the spatial comparison area did not provide a better outlier rejection, so we leave the box size at 3.

Create rmask file

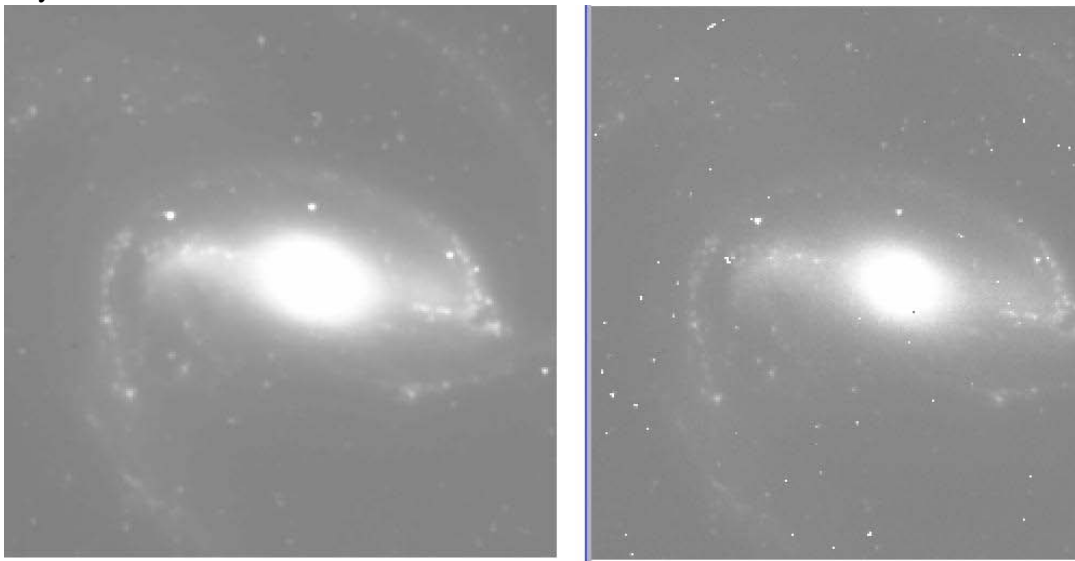
Add the following to your namelist file:

```
run_mosaic_covg = 1
run_mosaic_rmask = 1
```

```
&MOSAICRMASKIN
BOTTOM_THRESHOLD = 3,
TOP_THRESHOLD = 3,
BOX_BOTTOM_THRESHOLD = 6.0,
BOX_TOP_THRESHOLD = 6.0,
```

```
BOX_MIN_COVERAGE = 2,  
MIN_COVERAGE = 4,  
MAX_COVERAGE = 100,  
RM_THRESH = 0.5,  
REFINE_OUTLIER = 1,  
REFINE_OUTLIER_THRESH = 10  
&END
```

The dual outlier flagged pixels are used in the Rmask if an output pixel has been covered fewer than by MAX_COVERAGE input images. BOTTOM and TOP thresholds are not used in our example (they are only used for the temporal outlier method). The refinement parameters are not relevant for our example either. Decreasing the box thresholds to 3 produces too many outliers that do not correspond to real cosmic rays.



Final mosaic after cosmic ray rejection (left), one of the original CBCDs (right).

Reinterpolation and coadding

Add the following to your name list file:

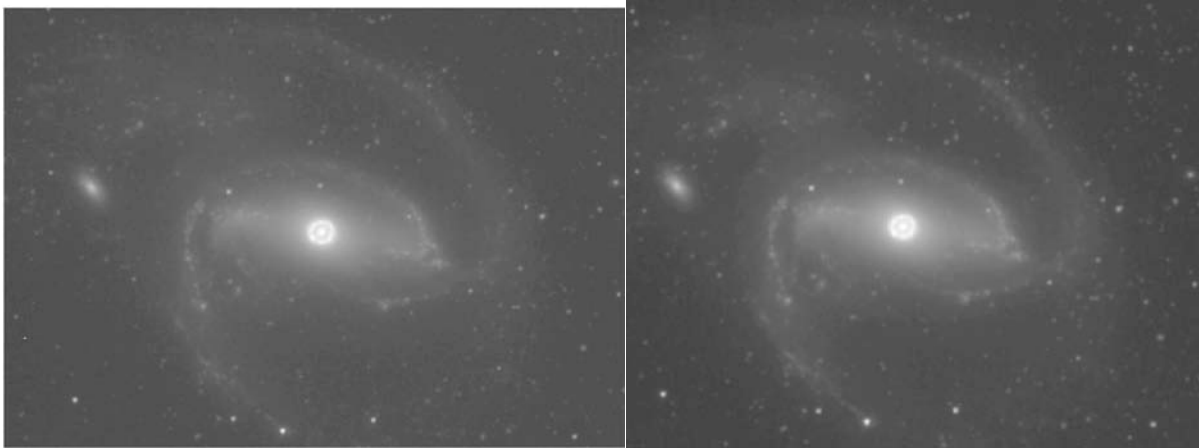
```
run_mosaic_reinterp = 1  
run_mosaic_coadder = 1  
run_mosaic_combiner = 1
```

```
&MOSAICCOADDIN  
TILEMAX_X = 1000,  
TILEMAX_Y = 1000,  
&END
```

Finally, run the mosaicer with

```
unix% mosaic.pl -n ngc1097ch1mos.nl
```

If you want to mosaic background-subtracted BCDs, set `RUN_MOSAIC_MEDFILTER = 1`



Pipeline-created mosaic in channel 2 (left), custom-tailored mosaic (made in this recipe; right).

CAVEATEMPTOR: More optimally constructed mosaics may possibly be constructed by using a different set of input parameters in the namelists. Use your own judgment and common sense.

Recipe 6. MOPEX: Making an IRAC Array Location Dependent Correction Mosaic

This is a recipe for making an IRAC array location dependent correction mosaic for a data mosaic made out of an IRAC dark field observation. We concentrate on the steps required to make the array correction mosaic and on how to use it on the fluxes, as the mosaicking steps are covered in Recipes 3-5 in this Cookbook, and the use of APEX is covered in other recipes.

Before starting it is a good idea to browse through the IRAC Instrument Handbook

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/iracinstrumenthandbook/>,

especially Section 4.5, which discusses the array location dependent effect, and the web pages at

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/locationcolor/>

which summarize the discussion of this effect and provide the per BCD correction files. In short, if you measure flux densities of blue sources (such as most stars), you should apply this correction to your measured fluxes, since the IRAC flatfield calibration is based on red sources (zodiacal light). You may want to apply this correction to individual (C)BCDs, especially when performing measurements on the individual frames, e.g., with PRF fitting. However, if you are performing aperture photometry on a well-dithered mosaic, you may want to make a correction mosaic from the individual correction images, built up exactly as your science mosaic, including outlier rejection. Then you can read off the correction for your targets from this mosaic. The dark field observation used in this recipe has a coverage of up to 18 per spatial position, although it is highly non-uniform (unlike a typical science observation would be). We use this dark field observation (from AORKEY 29841152) as a demonstration.

6.1 Requirements

You need to have MOPEX

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/mopex/>) installed to follow the instructions in this recipe.

If your data have been processed by version S18.0 or later of the IRAC pipeline, you may not need to run the artifact mitigation preprocessing, as explained in previous recipes, because corrected BCD (CBCD) data will be provided. However, it is always a good idea to look at the individual CBCD frames and see if there are any obvious remaining artifacts that could affect the data analysis down the line. In our case, although we see a couple of artifacts in a few frames, we proceed with mosaicking the CBCD frames.

6.2 Organizing Files and Directories

Before starting with this recipe, you should download the example data set using the Spitzer archive interface at

<http://irsa.ipac.caltech.edu/applications/Spitzer/SHA/>

The IRAC dark field AORKEY used in this recipe is 29841152. When downloading the data from the Spitzer archive, specify that you need the Level 1 data (Basic Calibrated Data; BCD). Since we are going to only demonstrate channel 1, you may just want to download channel 1 data. The BCDs are single images which have been run through the Spitzer online data reduction and calibration pipeline and the CBCDs have further gone through artifact mitigation. Create input files for the **cbcd.fits*, **cbunc.fits*, and **bimsk.fits* files. Ignore the first taken frame (*SPITZER_I1_29841152_0000_0000_4_cbcd.fits*) since this was taken in the HDR mode and the exposure time is different from the rest of the frames, which were 30-second exposures. Include the rest of the CBCD file names in this list and call this file, e.g., *"dark1_bcd.txt"*. In an analogous way, list the corresponding **cbunc.fits* files in a file called, e.g., *"dark1_unc.txt"*, and the **bimsk.fits* files in a file called, e.g., *"dark1_imask.txt"*. Each one of these three created files should list 18 file names, one per line.

Finally, you need to download the permanently bad pixel map (pmask):

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/pmask/>.

We use the pmask that was valid when our observations were made, as seen in the pmask fits file header keywords STRTDATE and ENDDATE. In our case it is *"sep07_ch1_bcd_pmask.fits"*.

6.3 Initial Setup

The first step is to specify which data you want reduced. You should edit the path to where you downloaded your data in the provided namelist file

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/dark_mosaic1.nl

in the following manner:

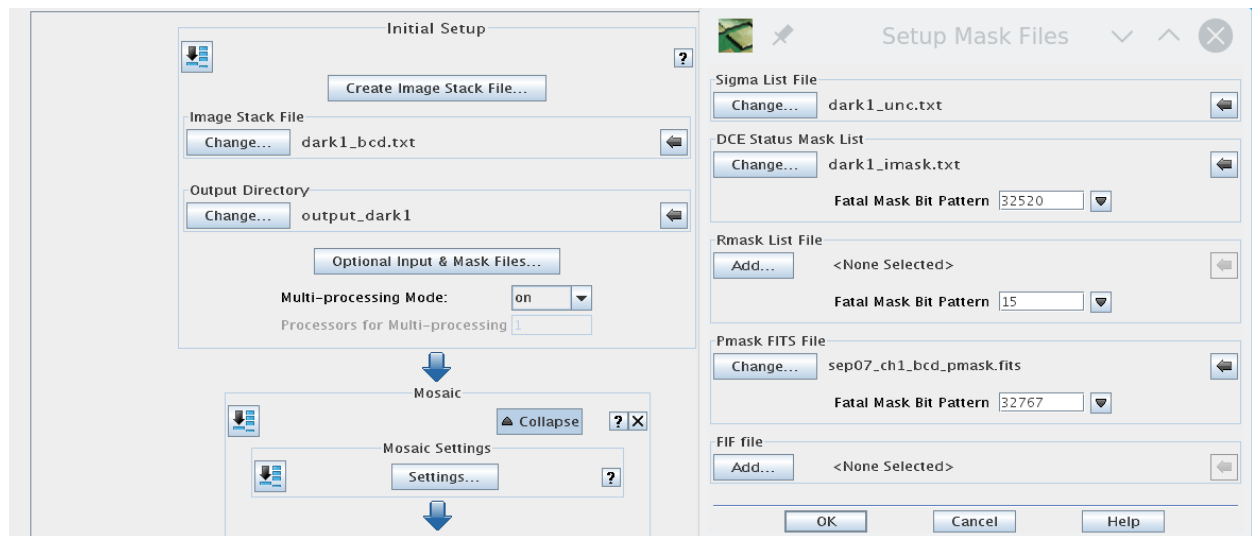
```
IMAGE_STACK_FILE_NAME = /my/data/directory/dark1_bcd.txt
SIGMALIST_FILE_NAME = /my/data/directory/dark1_unc.txt
DCE_STATUS_MASK_LIST = /my/data/directory/dark1_imask.txt
PMASK_FILE_NAME = /my/data/directory/sep07_ch1_bcd_pmask.fits
ARRAY_CORR_IMAGE = /my/data/directory/ch1_photcorr_rj.fits
```

```
ARRAY_PIXAREA_IMAGE = /my/data/directory/ch1relpixarea.fits
OUTPUT_DIR = /my/data/directory/output_dark1
```

Replace `/my/data/directory/` with your local directory path where the files are located. For “Array Correction Frame” the correct file in ch 1 is “ch1_photcorr_rj.fits” for the cryogenic mission, as is the case for the data used in this example, and “ch1_photcorr_ap_5.fits” for warm mission data.

You may edit the supplied namelist file first, and then read it into MOPEX GUI by clicking "File->Read Name List...". Or you may just read the namelist file into the MOPEX GUI and then edit the input files. Do this by clicking "Add..." under "Image Stack File" and specifying the path to the file containing the list of IRAC Channel 1 CBCD image files that you created earlier ("*dark1_bcd.txt*"). Then click "Add..." under "Output Directory" and specify where the intermediate and final data products should be saved, e.g., "*output_dark1*." Next, click "Optional Input & Mask Files", and then click "Add..." under "Sigma List File" and specify the list of cbunc files created earlier ("*dark1_unc.txt*"). Then click "Add..." under "DCE Status Mask List" and specify the list of bad pixel map files created earlier ("*dark1_imask.txt*").

Specify the Fatal Mask Bit Pattern (32520 is recommended as of March 2010, it uses bits 3, 8-14 of the imasks). Specify Rmask fatal bit pattern (15, we use all the outlier rejection methods for demonstration purposes in this recipe, but you may be OK by using just one or two of them for your data, as discussed in other recipes in this Cookbook). Specify also the Fatal Mask Bit Pattern for pmask, which is 32767, so using all the bits. The FIF file does not need to be specified. Once you are finished specifying the input parameters click "OK" to return to the pipeline view.



Initial mosaic settings.

6.4 Channel 1 Mosaicking Settings

The "Mosaic Settings" dialog box sets several global parameters related to image combination and temporary files. First, the pixel size is set, to make mosaics with 0.6 arcsec x 0.6 arcsec pixels (which is now the default for IRAC pipeline mosaics as well).

Next you need to set the array location correction images and pixel area images for channel 1. The array location dependent photometric correction image corrects the photometry for the fact that the flat-fielding was made with the help of a red source (zodiacal light) and is thus not correct for blue sources, such as most stars. The pixel area images are simply images of the pixel solid angles relative to the center pixel, which are used for image distortion correction (or its canceling, as the case is here). Download these from

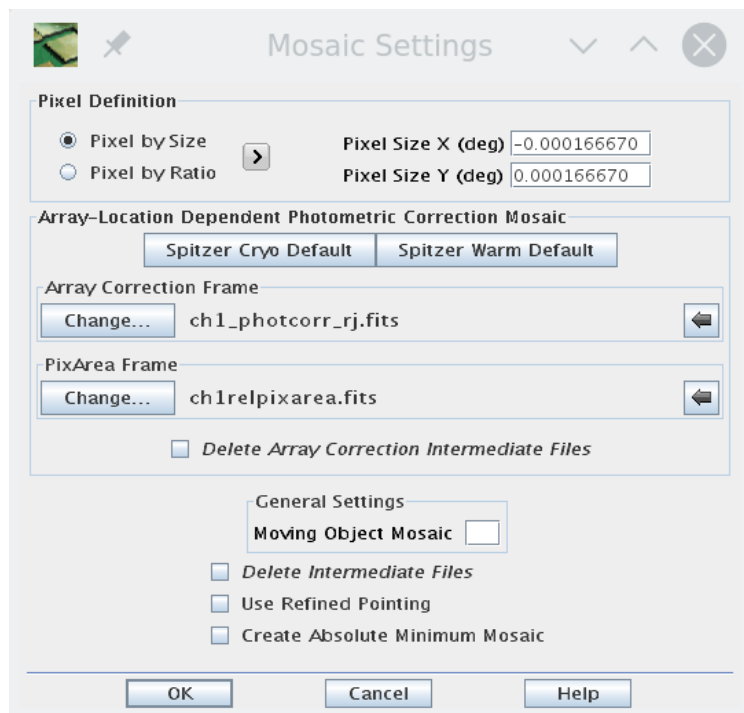
<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/locationcolor/>

and

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/solidangles/>

Specify the channel 1 images (for "Array Correction Frame" the correct file is "*ch1_photcorr_rj.fits*" and for "PixArea Frame" it is "*ch1relpixarea.fits*"). Because the correction images already take IRAC image distortion into account, MOPEX will need to divide the array correction images by the Pixel Solid Angle images, because MOPEX corrects for the image distortion. Otherwise the image distortion would be corrected twice.

You may leave the rest of the options unselected in the "Mosaic Settings" dialog and just click "OK".



Mosaic Settings dialog.

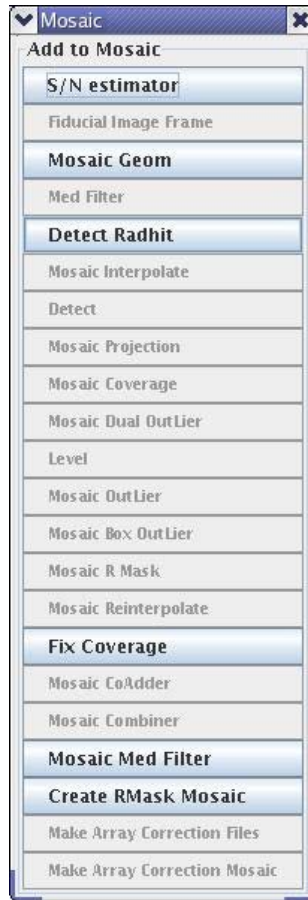
6.5 Mosaicking

The various modules for mosaicking have been discussed in other recipes in this Cookbook, specifically recipes 4, 5, and 6. Here we just use the template values in the MOPEX GUI for them.

6.6 Array Location Correction in MOPEX

To make the array location correction images, you need to add two modules to the end of the mosaicking pipeline, "Make Array Correction Files" and "Make Array Correction Mosaic". If you use the provided namelist, they are already added. If you are creating your own namelist in GUI, you can click on the

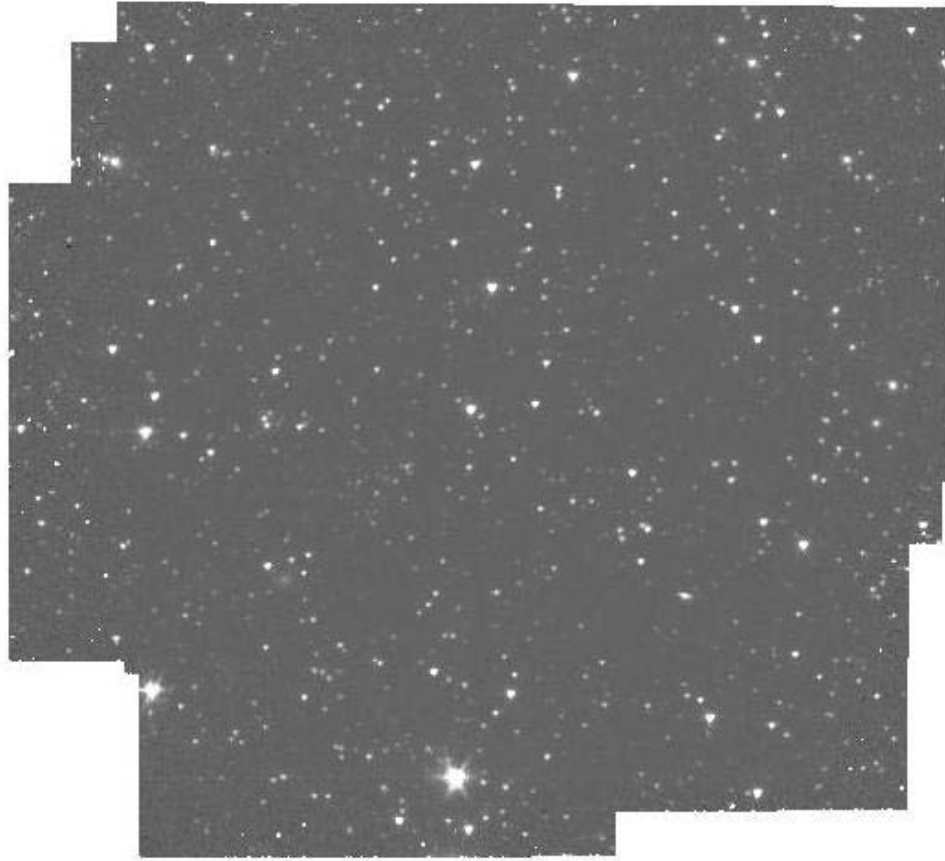
mosaic module list that is displayed in a separate window (to the right of the module flow depiction). Just click on the "Make Array Correction Files" and "Make Array Correction Mosaic" to add them into the flow. There are no parameters to be specified for these two modules (but make sure you specified the input array correction frame file and the pixel solid angle image, as discussed above).



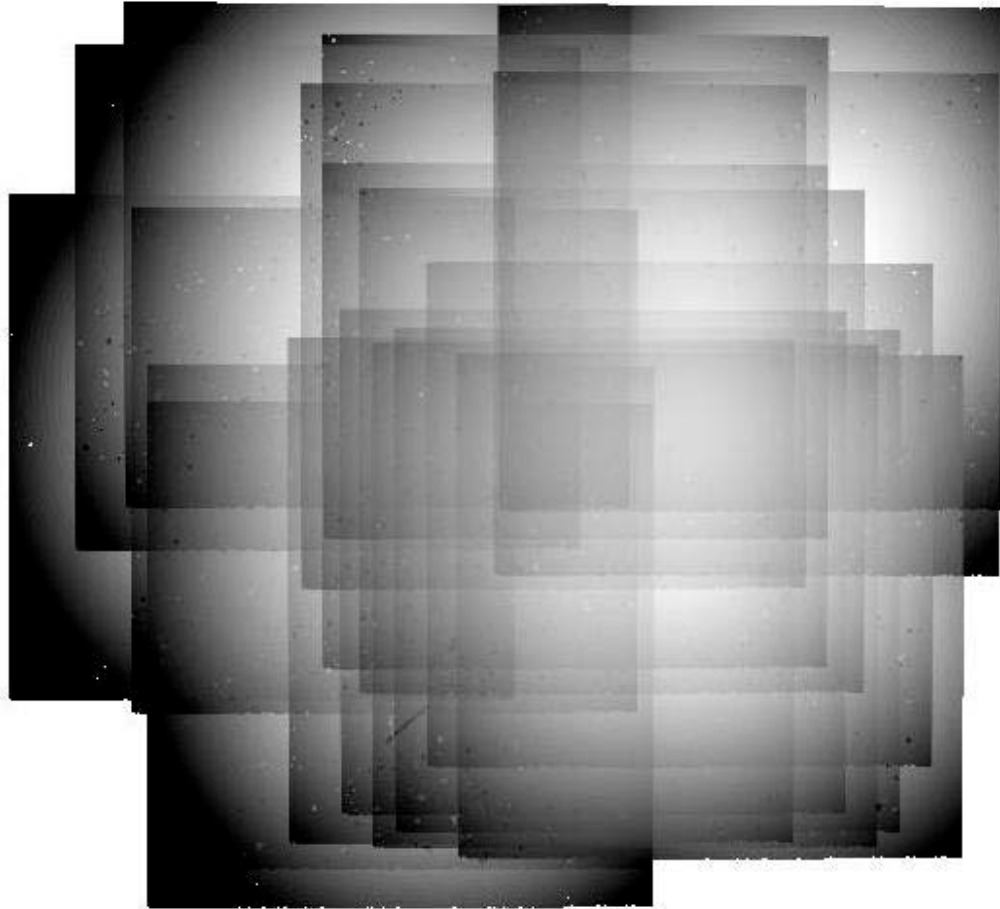
Mosaic module list window from which new modules can be inserted into the processing flow.

6.7 Making the Array Location Correction Mosaic

Now when you press the green "play" button at the top of MOPEX GUI, it will create both the science image mosaic and the array location correction mosaic for you. Both will be placed under the output directory in a subdirectory called "Combine-mosaic". The name of the science mosaic is "mosaic.fits" and the name of the array location correction mosaic is "mosaic_arraycorr.fits". Images are shown below.



Science mosaic for the darkfield observation.



Array location dependent correction image mosaic for the dark field.

6.8 Performing Photometry on the Bright Point Sources

We use APEX to perform photometry on some of the brightest point sources in the created science mosaic "*mosaic.fits*". We use the template file for performing user list single file frame aperture photometry in MOPEX GUI, and edit this file slightly. See the

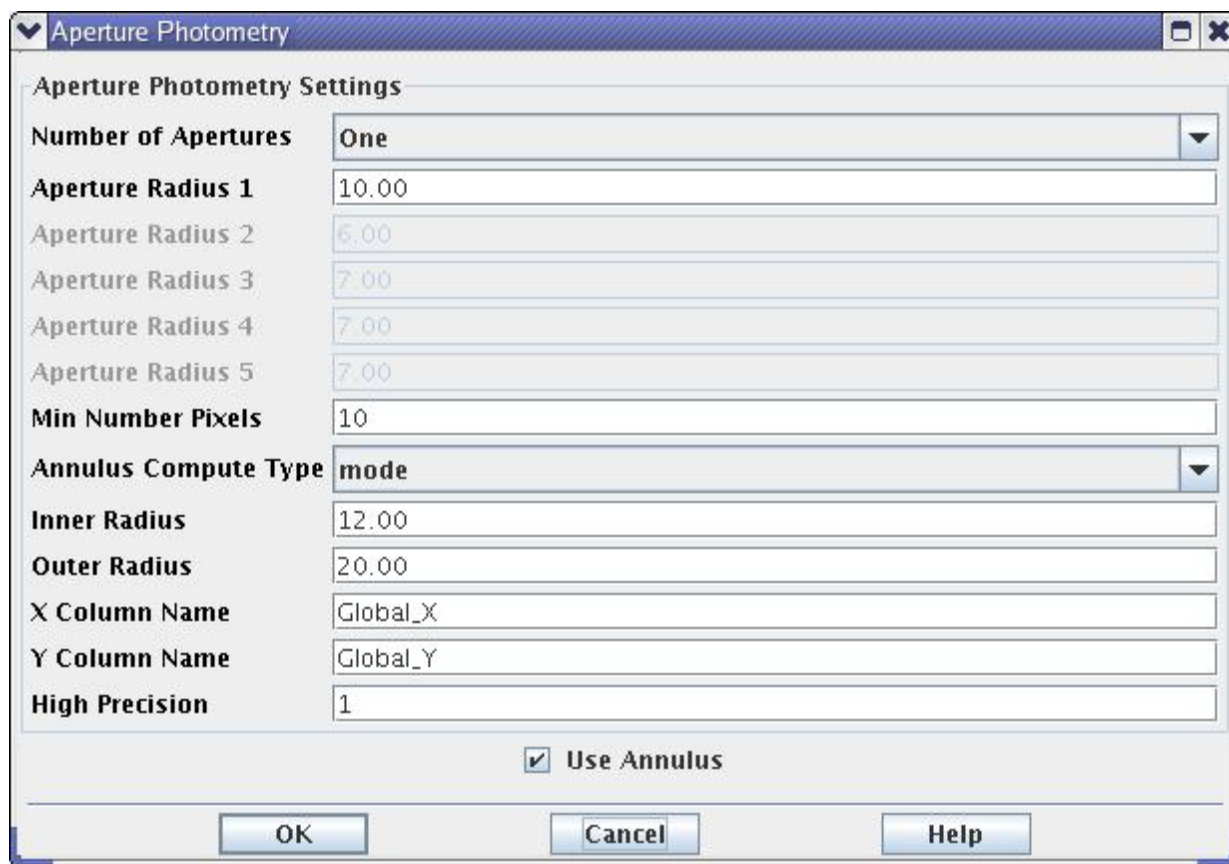
http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/dark_apex1.nl

namelist. You can load this namelist into MOPEX GUI by selecting "File->Read Name List..." and then selecting "Apex User List single frame." Edit the input file paths and names. The INPUT_FILE_NAME should be the mosaic created above ("*mosaic.fits*"), the SIGMAFILE should be the mosaic uncertainty image, to be found in the subdirectory "Combine-mosaic" ("*mosaic_unc.fits*") and the COVERAGE_MAP should be in the same subdirectory ("*mosaic_cov.fits*"). The INPUT_USER_LIST specifies the positions of the centroids of the bright stars, and you can use the supplied file

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/dark1_star_list.nl

or create one for yourself by measuring the centers of star positions with your favorite software (remember that APEX uses the convention where the center of the lower left pixel is 1.0, 1.0). Convert the pixel coordinates into RA and Dec coordinates using the header information in the mosaic (CRVAL,CRPIX, CD matrix values).

Please check the settings under the Aperture Photometry (or under &APERTURE in the provided *"dark_apex1.nl"* file) parameters as well, as we use the 10.0 pixel radius and the 12.0 - 20.0 pixel background annulus, the same as those that are used for IRAC calibration stars, so no aperture correction is needed. We only use this one radius setting, so N_Apertures = 1. These are set in the given namelist *"dark_apex1.nl"*, but if you are building your own namelist, you will need to change these values by hand. After all these are set, you should be able to run APEX by pressing the green "Go" button on the top pane of MOPEX GUI. The results will be in the specified output directory (in the case of the template, *"output_apex"*, in the file named *"dark1_mosaic_aperture.tbl"*, if we used *"dark1_mosaic.fits"* as the name of the input mosaic file, otherwise the resulting file will be called the "name_of_your_input_mosaic_file_aperture.tbl"). At the bottom of this file are the aperture measurements. The columns are the source id, x and y coordinates in pixels, aperture flux density in micro-Jy, bad pixel number (if any), uncertainty for aperture flux density, and background/pixel estimate that was subtracted, and the RA and Dec decimal coordinates).



Aperture photometry settings window in MOPEX GUI.

6.9 Correcting Flux Densities with the Array Location Correction Mosaic

Now the corrections for flux densities can be looked up in the array location correction mosaic at the coordinates of the sources mentioned in the **aperture.tbl* file. Display the correction mosaic fits file, and place the cursor on the coordinates of the source mentioned in the **aperture.tbl* file, and read off the pixel value at that location. Then multiply the source flux by this value. For this recipe, we get the following corrections:

X-coord	Y-coord	Original Flux (mJy)	Correction	Corrected Flux (mJy)
522.05	506.83	1.8767	0.999475	1.8757
545.12	639.61	1.9935	1.00141	1.9963

526.55	774.00	1.0234	0.997992	1.0214
633.37	870.34	1.6323	0.995319	1.6247
167.46	481.66	3.3302	0.985843	3.2831
885.00	358.66	1.7004	1.00293	1.7054
782.90	170.20	1.4806	0.997420	1.4768

Remember that for your science targets, you may need to apply other corrections, such as the color correction (IRAC Instrument Handbook 4.4), or aperture corrections if you are not using the IRAC calibration star aperture and background annulus.

Recipe 7. MOPEX/APEX: Point Source Photometry from IRAC Images

This recipe illustrates how to perform point-source photometry from IRAC image data with a relatively uniform background. We show how to use APEX, a photometry package that is part of [MOPEX](#), the MOsaicker and Point source EXtractor. APEX will be used in its multiframe mode, performing detection on coadded data, but point-source fitting photometry on the individual frames. (Aperture photometry is also available, and this is done on the mosaic.) For this example, we will show how you can run all steps in one continuous flow, starting with the Overlap module to correct background levels, and ending with the APEX QA (Quality Assessment) module to examine the quality of the PSF fits. We will demonstrate this using the MOPEX GUI, and we recommend that you also refer to Recipe 4, on using the GUI to mosaic IRAC data. In this recipe, we are analyzing channel 1 (3.6 micron) data from the Galactic First Look Survey (GFLS;

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/spitzermission/observingprograms/firstlooksurvey/>).

Note: With MOPEX release 18.4, default templates will make mosaics with 0.6" pixels, like the PBCD mosaics. This example has been updated to reflect this.

7.1 Requirements

You must have the Spitzer software package MOPEX installed in order to follow along with this recipe (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/mopex/>). This demonstration was originally tested using MOPEX 18.4.9. Use the Spitzer Heritage Archive (<http://irsa.ipac.caltech.edu/applications/Spitzer/SHA/>) to download the BCD and post-BCD data for AOR 4960000 for this example. See Recipe 1 and Recipe 2 for examples on downloading the data. You will need to request Level 1 (BCD) data and Level 2 + ancillary data (PBCD).

7.2 Have a quick look at your data

It is always a good idea to have a quick look at your data before you proceed with a detailed reduction. The Spitzer pipeline produces a mosaic for just this purpose. To examine it, `cd` into the Post-BCD area of your data directory:

```
cd /where/you/unpacked_data/r4960000/ch1/pbcd/
```

The observations in this particular AOR used the IRAC High Dynamic Range mode for imaging. Hence, there are data for both a short exposure and a long exposure.

Using `ds9` or any other viewer of your choice, examine the channel 1 mosaics for the two sets of exposures. The long exposure mosaic is called `SPITZER_I1_4960000_0000_8_E8349797_maic.fits`, and the short exposure mosaic is called `SPITZER_I1_4960000_0000_8_A42451142_maics.fits`. (You can find a summary of the IRAC filenaming convention in the [IRAC Instrument Handbook](#).) As you can see, the most recent version of the IRAC pipeline does a pretty good job of producing a mosaic that has a relatively smooth background, etc.

However, you should just about always produce a mosaic yourself, using [MOPEX](#), from the original BCD data. This is not the purpose of this particular recipe, although a science-quality mosaic will be a byproduct of the processing in this recipe.

In the example below, we will be concentrating on point source extraction from the long exposure BCD data. An analogous procedure can be done to perform point source extraction on the short-exposure data, in order to measure the brightnesses of stars that are saturated, or nearly saturated, in the long-exposure data.

7.3 Artifact Mitigation

IRAC BCD data can suffer from several different artifacts, including:

- Stray light
- Muxstripe
- Muxbleed
- Column Pulldown
- Jailbars

As of version S17.0 of the Spitzer IRAC pipeline, artifact-mitigated images and their associated uncertainty images (`*cbcd.fits` and `*cbunc.fits`) are now available in the archive. These images include corrections for column pulldown and banding, induced by bright sources in the images. The corrections are empirical fits to the BCDs and may not always improve the data quality. The standard BCD files (`*_bcd.fits`) remain available in the archive. The mosaics (pipeline post-BCD products) are now created from the `*cbcd.fits` images.

In addition to inspecting the post-BCD mosaic, you should also generally inspect the individual CBCD images, to determine whether remaining artifacts can be seen in them and then perform additional artifact mitigation, if necessary.

The data we are considering in this recipe were processed with pipeline version S18.18, as can be seen in the file

```
/where/you/unpacked_data/r4960000/QualityAnalysis_4960000.README
```

so, some artifact mitigation has already been performed. For this reason, we will use the `*cbcd.fits` files. We will also assume the corrections were good, and use the original uncertainty files, `*bunc.fits`.

7.4 Set up files in preparation to run MOPEX

Before we can run MOPEX to compute the overlap correction, mosaic the data, and perform point source extraction, we need to do a little preparation.

- Make a list of the CBCD (**cbcd.fits*), uncertainty (**bunc.fits*), and bad pixel mask (**bimsk.fits*) files in the directory */where/you/unpacked_data/*. For this recipe, the lists should include only the odd-numbered frames, which have long exposure times (header keyword `HDRFRAME = long`). For your convenience, we provide a shortened list of image files for the demonstration here:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/gfls_cbcd.txt

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/gfls_bunc.txt

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/gfls_bimsk.txt

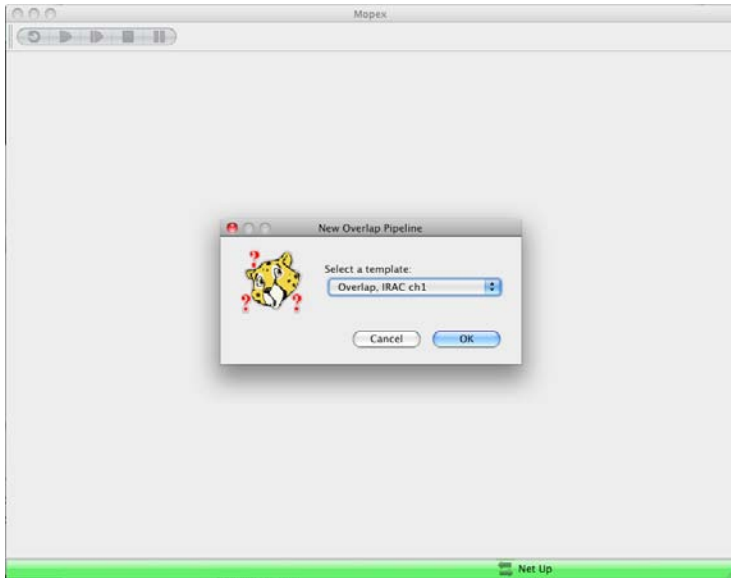
Remember to edit these files so that each line contains the full paths to the data on your computer.

We can run each of the steps below in sequence, or we can load each of the individual pipelines into a flow in MOPEX and run the flow all the way through. We choose to do the latter here. However, you could certainly run as a flow each pipeline separately, adding a new pipeline after running a previous one.

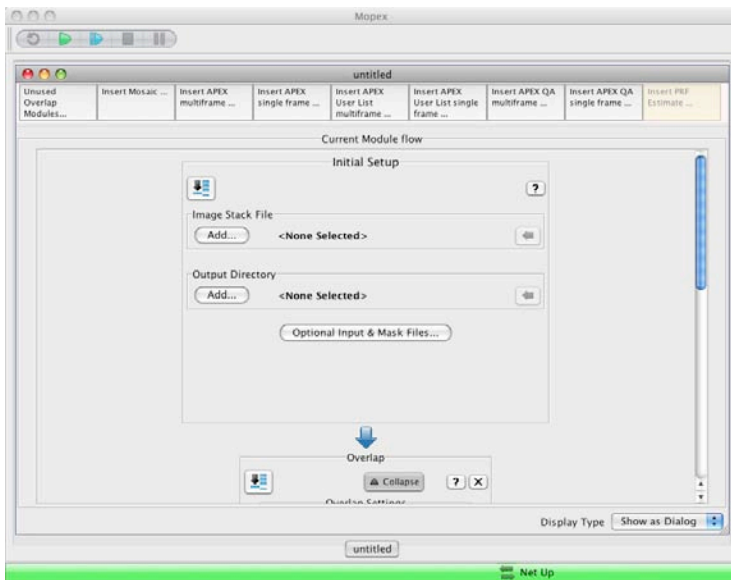
7.5 Compute the overlap correction

First, we must match the backgrounds in overlapping IRAC frames. The Overlap Pipeline in [MOPEX](#) does this by calculating an additive correction for each CBCD image, to bring them to a common background level. Follow these steps to run the overlap pipeline on the corrected BCDs. But do not run the flow yet -- that will come later.

1. Start up the MOPEX GUI.
2. Under the File menu, select a New Overlap Pipeline. Click on "File"-->"New Overlap Pipeline..."-->"Overlap, IRAC ch1"-->click "OK".



You will see the previously empty MOPEX window fill up with the Overlap flow.



3. Set parameters in MOPEX Overlap Flow.

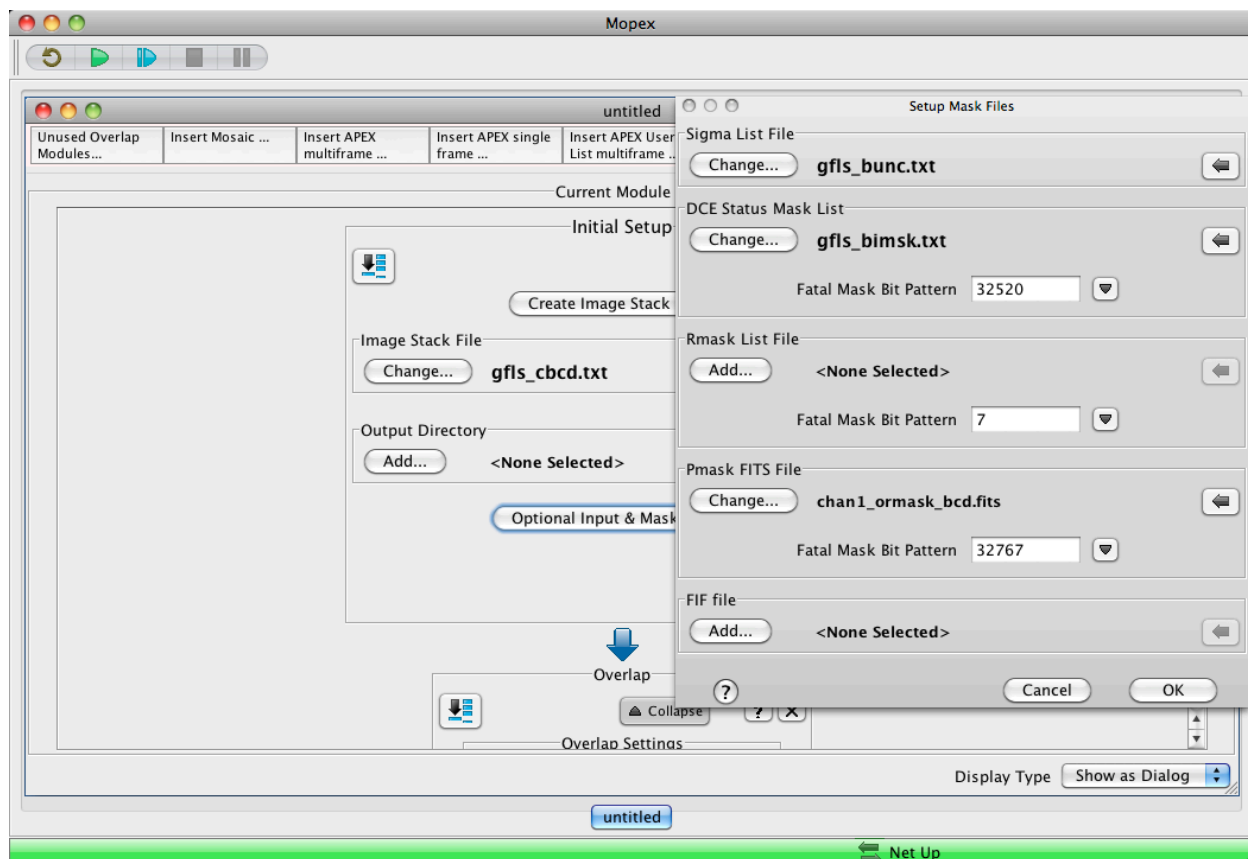
Below is the minimum set of parameters that should be set in running overlap. A full description of each of the parameters is provided in the [MOPEX User's Guide](#) and is also discussed in the IRAC COSMOS Recipe (Recipe 3).

- a. Image Stack File should be the *gfls_cbcd.txt* file you downloaded and edited previously.
- b. Output Directory is a directory you create, e.g., */where/you/unpacked_data/ch1/bcd/output/*.

c. Optional Input & Mask Files -- Sigma List File should be the *gfls_bunc.txt* file you downloaded and edited previously.

d. Optional Input & Mask Files -- DCE Status Mask List should be the *gfls_bimsk.txt* file you downloaded and edited previously.

e. Pmask FITS file – The GUI template will load a default Pmask (permanent bad pixel mask) file that should be good for most purposes. Date-specific Pmasks for IRAC are available in the *cal/* subdirectory where you installed [MOPEX](#), e.g., */your_home_directory/mopex/cal/* (assuming [MOPEX](#) is installed in your home directory). You can check that you have the latest versions as follows. A tarball of the Pmasks can be found at <http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/pmask/>. You can unpack the tarball in *cal/*. (This will overwrite any potentially older versions of these files.) To decide which Pmask to use, look at the date your BCDs were observed. From the keyword DATE_OBS in the fits header of the BCD files, we find that these data were taken on December 6, 2003. Look in the *cal/* subdirectory for the Pmasks closest in time to the observation date; in this case, those in the *mar04/* subdirectory. To verify that the channel 1 Pmask file here, i.e., the file */your_home_directory/mopex/cal/mar04/mar04_ch1_bcd_pmask.fits* is the appropriate one, you can look at its header for the keywords STRTDATE and ENDDATE to see the date range for which the Pmask is applicable. Since this date range is August 1, 2003 to May 2, 2004, we see that this is the right one.



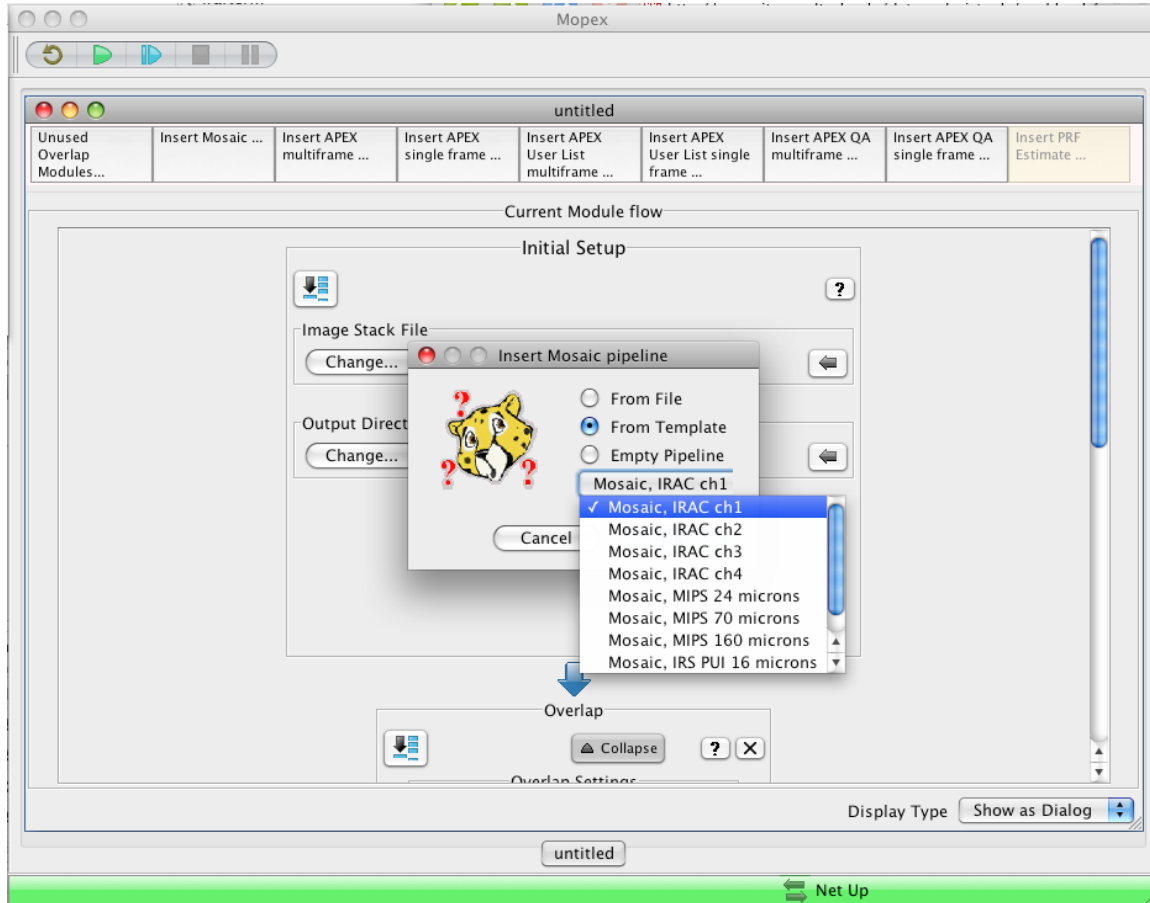
We could make a quick mosaic within Overlap to see how good the corrections were. But we will skip that step and make a better mosaic by inserting a Mosaic step. It will use the overlap-corrected frames automatically.

7.6 Produce a Science Mosaic

Now that we have corrected the BCDs for mismatched backgrounds, the next step is to use the corrected BCDs to produce a new mosaic.

1. Insert a Mosaic Pipeline into the flow.

On the MOPEX GUI, click where it says, “Insert Mosaic” In the dialog box, select “From Template,” and “Mosaic, IRAC ch1,” and click “OK.”



By-and-large, the default parameters for this pipeline can be used. As described in the "Basic Concepts: Outlier Detection" section of the online MOPEX manual, MOPEX has four algorithms for identifying bad pixels: single-frame, multiframe temporal, dual outlier, and box outlier. The single-frame outlier method is not normally needed. For low coverage data like ours (the AOR we have chosen has a coverage ~ 4),

you should use either dual outlier or box outlier, or both. The default IRAC Mosaic template uses all three main methods. Since dual and box are both present, we can leave this as it is.

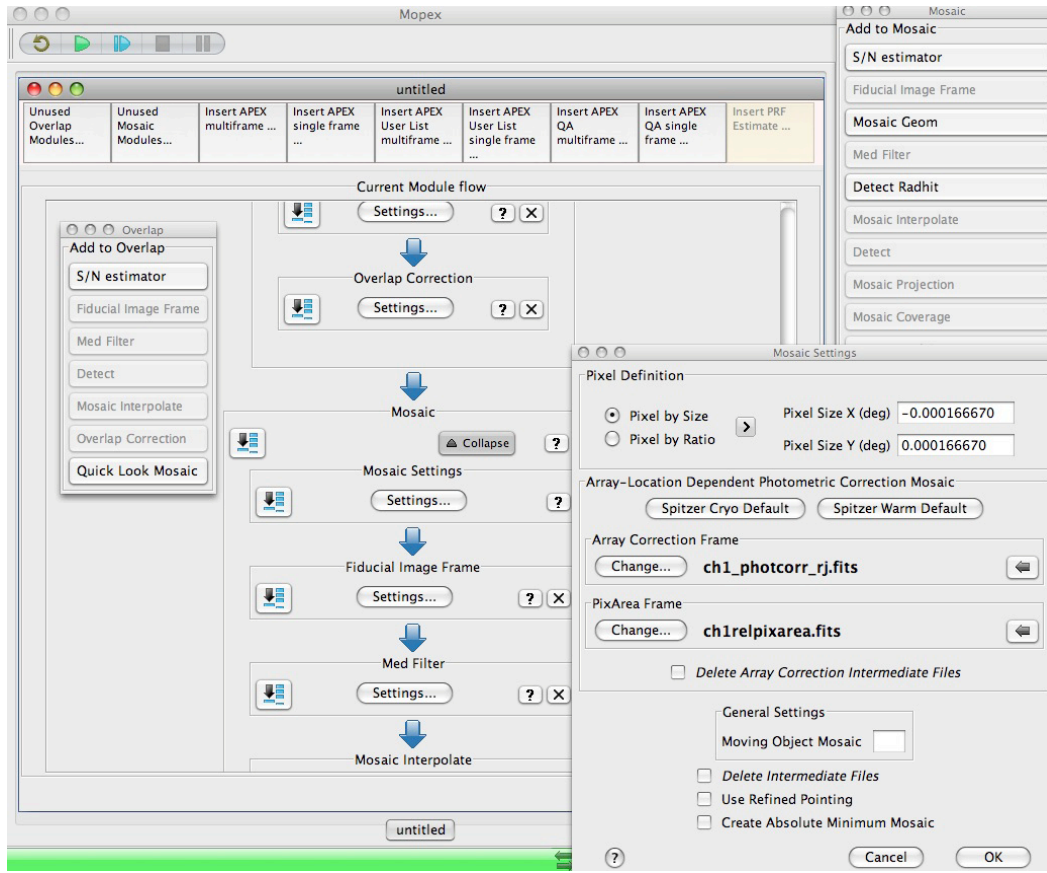
Additionally, as described in the [MOPEX User's Guide](#), four interpolation schemes are implemented: Default, Drizzle, Grid and Cubic. General guidelines for choosing the interpolation scheme appropriate for your data set are provided in the [Helpdesk](#) knowledgebase: <http://sohel2.ipac.caltech.edu/support/index.php?/Knowledgebase/Article/View/68>. A rule of thumb is that if you have a coverage of 10 or less (as we do), you will probably want to use the default interpolation scheme.

If you have higher coverage data than the low coverage data we are analyzing here, you may wish to use a different outlier rejection scheme and a different interpolation method. For a medium-deep coverage example, see the IRAC COSMOS recipe (Recipe 3).

2. Array-Location Dependent Photometric Correction

As discussed in the IRAC Instrument Handbook (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/iracinstrumenthandbook/>), the IRAC flat-field is derived by imaging the high surface brightness zodiacal background, and the spectrum of the zodiacal background peaks redward of the IRAC filters. However, the spectral energy distribution of many sources, particularly stars, have color temperatures that peak blueward of the IRAC filters. A variation also exists in the effective filter bandpass of IRAC as a function of angle of incidence, which in turn depends on the exact position of an object on the array. Photometry of blue sources like stars needs a correction factor. There are two ways to do this. The first, more exact, method is to correct the (C)BCD frames before making the mosaic. The second, approximate, method is to create a mosaic of correction factors, and multiply extracted fluxes by the value in the correction mosaic at the same position as the object. The GUI is currently set up to create the correction mosaic for the second method.

You can make the correction mosaic easily within the Mosaic pipeline using the GUI, by clicking on the “Settings” button in the Mosaic Settings module of the pipeline. In the resulting dialog box, if you click on “Spitzer Cryo Default,” under “Array Location Dependent Photometric Correction Mosaic,” the appropriate files that you need to apply this correction (in this case, *chl_photcorr_rj.fits* and *chlrelpixarea.fits*) will be input. These files can be found in *cal/*. Then from the Add to Mosaic window click on "Make Array Correction Files" and "Make Array Correction Mosaic" to add these to the flow. The correction mosaic will go into the same output directory as the normal mosaic. Refer to Recipe 6 for more detailed instructions on how to build an array location dependent mosaic.

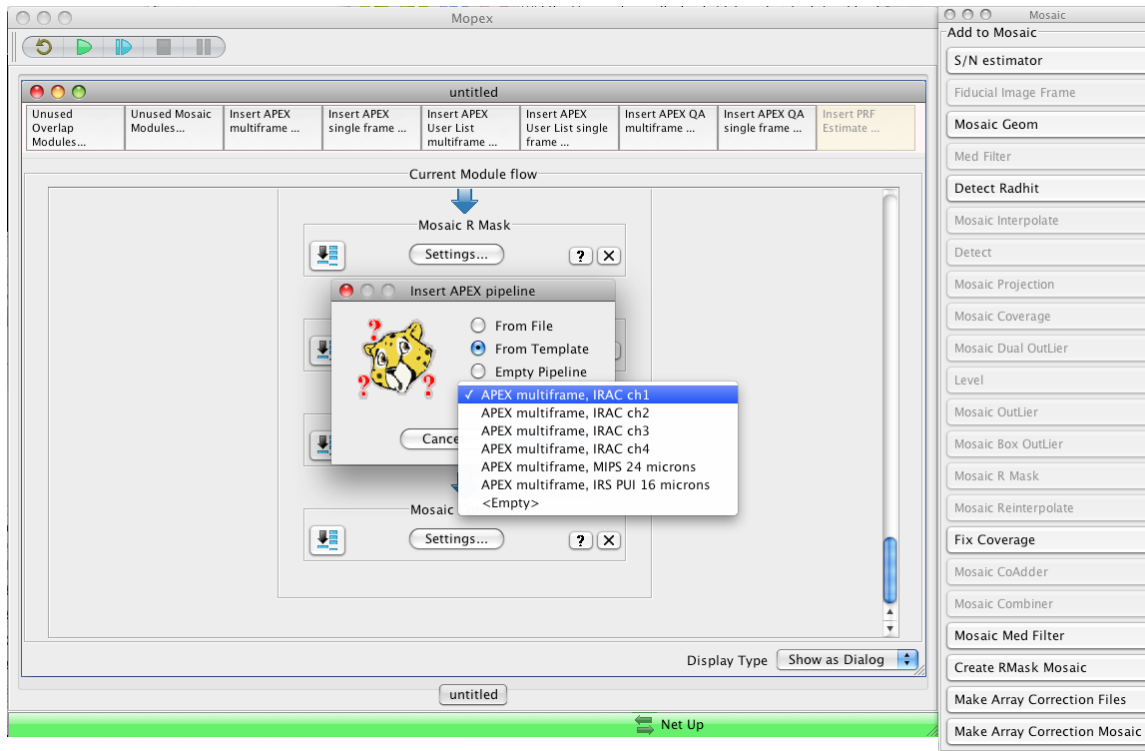


7.7 Perform APEX Multiframe photometry

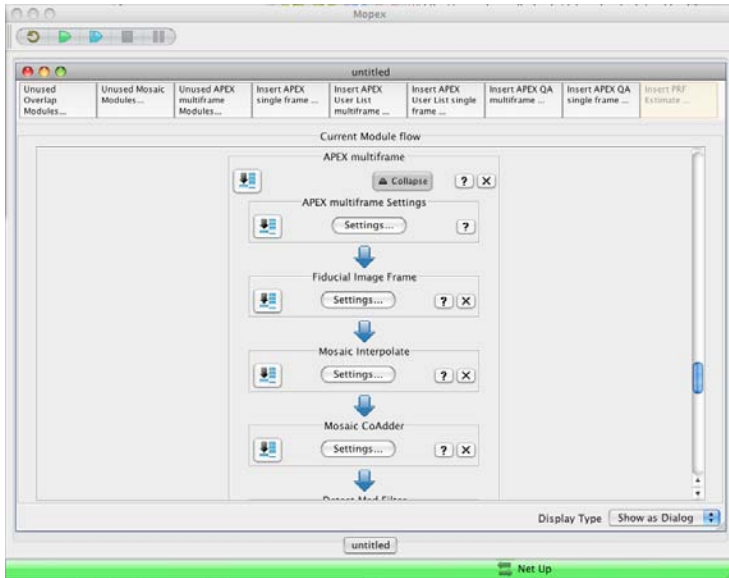
As part of this flow, we are constructing a science-grade mosaic. But for IRAC, point-source fitting needs to be done on the individual (C)BCD frames, not on the mosaic. (Aperture photometry will be done on the mosaic.) For the point-source fitting, we will use a map of the spatially-variable PRF for the highest accuracy.

1. Insert APEX Multiframe pipeline

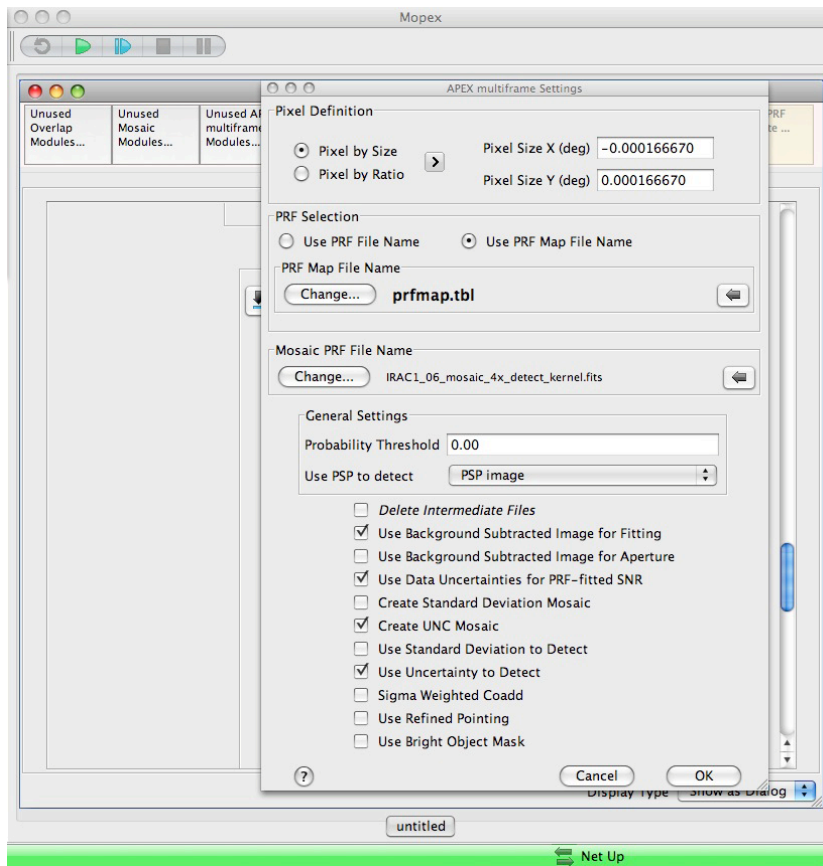
On the MOPEX GUI, click where it says, “Insert APEX multiframe....” In the dialog box, select “From Template,” and “APEX multiframe, IRAC ch1,” and click “OK.”



You will see the APEX Multiframe pipeline inserted into the overall flow.



2. Point APEX Multiframe to a PRF Map



The MOPEX GUI by default has the PRF selection, under the APEX Multiframe settings dialog, as the file *apex_sh_IRAC1_coll29_row129_x100.fits*, which is located in the */your_home_directory/mopex/cal/* subdirectory. Although you can run APEX with just this PRF, relevant to the IRAC frame pixel center,

we recommend using the whole set of PRFs, in a PRF Map, as this noticeably improves the quality of the PRF fitting for sources outside of the central region of the arrays. The set of PRFs for channel 1 can be found at

http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/psfprf/070131_prfs_for_apex_v080827_ch1.tar.gz and the corresponding PRF Map file can be found at

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/psfprf/prfmap.tbl>. The gzipped tarball of PRF FITS files can be unpacked in the `/your_home_directory/mopex/cal/` subdirectory. Edit the `prfmap.tbl` file so that the prf file names contain the full directory path. For example, the line for `PRF_Filename_1` might look like:

```
\char PRF_Filename_1 =  
/Applications/mopex/cal/070131_prfs_for_apex_v080827_ch1/apex_sh_IRAC1  
_col025_row025_x100.fits
```

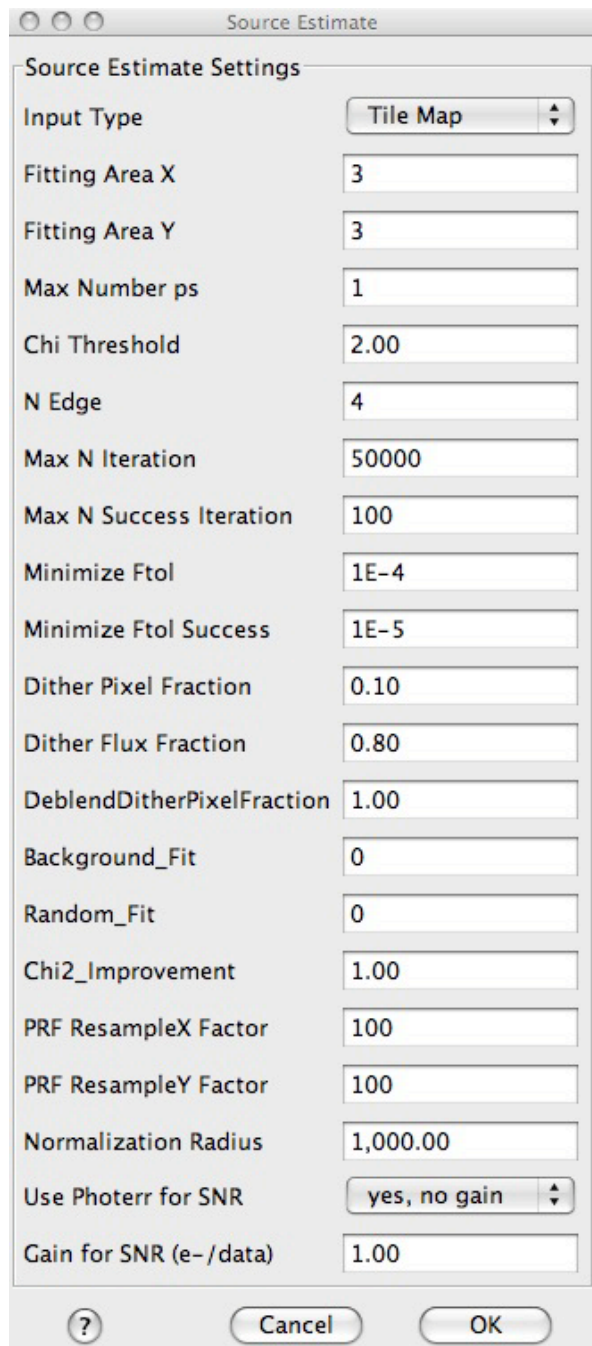
In APEX Multiframe Settings, select "Use PRF Map File Name" and choose `prfmap.tbl`. In "Mosaic PRF File Name" ensure the following file from the `cal/` directory is present:

`IRAC1_06_mosaic_4x_detect_kernel.fits`. In the default template, APEX needs this to detect on the 0.6" pixel mosaic.

The rest of the default settings should be acceptable for the present purposes. But the default settings for APEX will create a crude mosaic (no outlier rejection). Since we already are creating a mosaic, we need to remove some modules from the APEX flow.

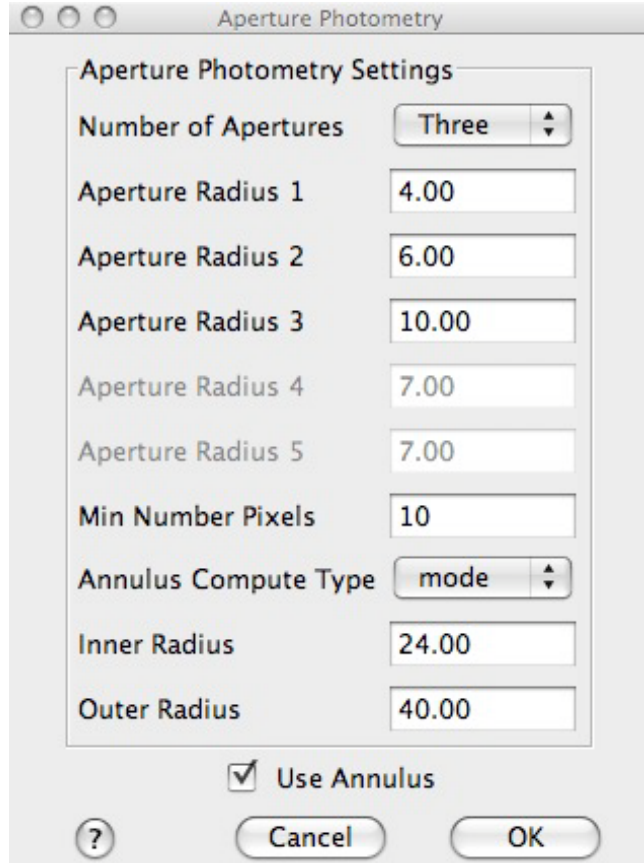
Simply click on the "X" to the far right in the modules, "Fiducial Image Frame," "Mosaic Interpolate", "Mosaic CoAdder," and "Mosaic Combiner." This will remove these modules from the flow.

3. Check the Source Estimate Settings



Most, if not all, of the default settings for the Source Estimate module are fine to use without modification. Check these settings by clicking on “Settings” in the module window. In particular, if you are performing PRF fitting with the “100×”-sampled PRFs, either the single, center-frame PRF or the PRF Map (recommended), you should be sure that the “Normalization Radius” is set to the value “1,000.00.” (The “standard aperture” for the IRAC absolute calibration is 10 pixels; these model PRFs have been 100-times-sampled, hence, 10 pixels \times 100 = 1,000 pixels.) This will ensure that the PRF flux normalization is matched to the IRAC calibration standard radius.

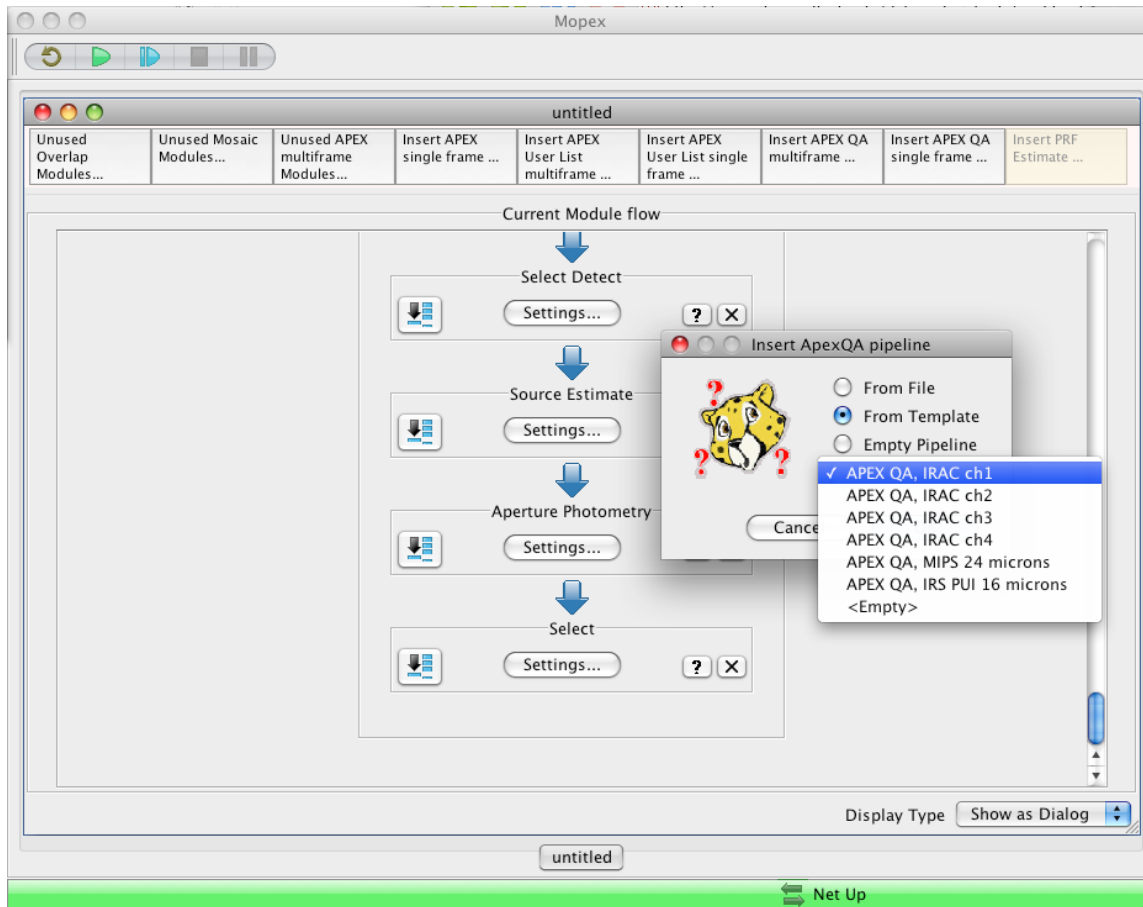
5. Examine the Settings in the Aperture Photometry module.



Click on the Settings button in the Aperture Photometry module. Note the default aperture radii of 4.0, 6.0, 10.0 in 0.6" mosaic pixels correspond to BCD image apertures (2.0, 3.0, 5.0 BCD pixels) for which aperture corrections are available in the IRAC Instrument Handbook. A choice of 20.0 (10.0), together with the annulus settings of Inner Radius of 24 (12) and Outer Radius of 40 (20), would correspond to the standard aperture used for IRAC flux calibration. If desired, add this aperture. However for typical sources in typical fields, this is too large of an aperture for science purposes.

7.8 Perform Quality Assessment of the PRF Fitting

1. Insert the APEX QA Multiframe pipeline



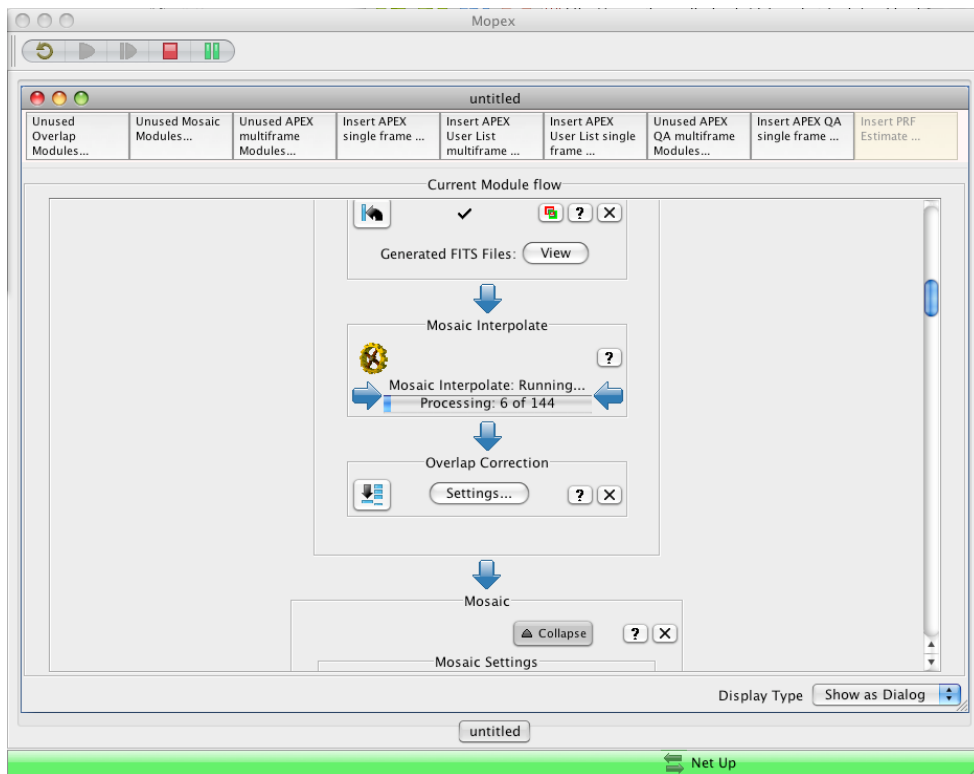
To assess the quality of the PRF fitting on the BCD frames performed in the APEX Multiframe pipeline, we can use the APEX QA Multiframe pipeline. This module can subtract fitted PSFs from the BCD frames and optionally mosaic those residual frames. It will reveal if some sources have been over- or under-subtracted, e.g. due to difficulty fitting a close blend of sources.

Above the flow, click on the box "Insert APEX QA multiframe...". In the dialog box, select "From Template" and "APEX QA, IRAC ch1" and click "OK".

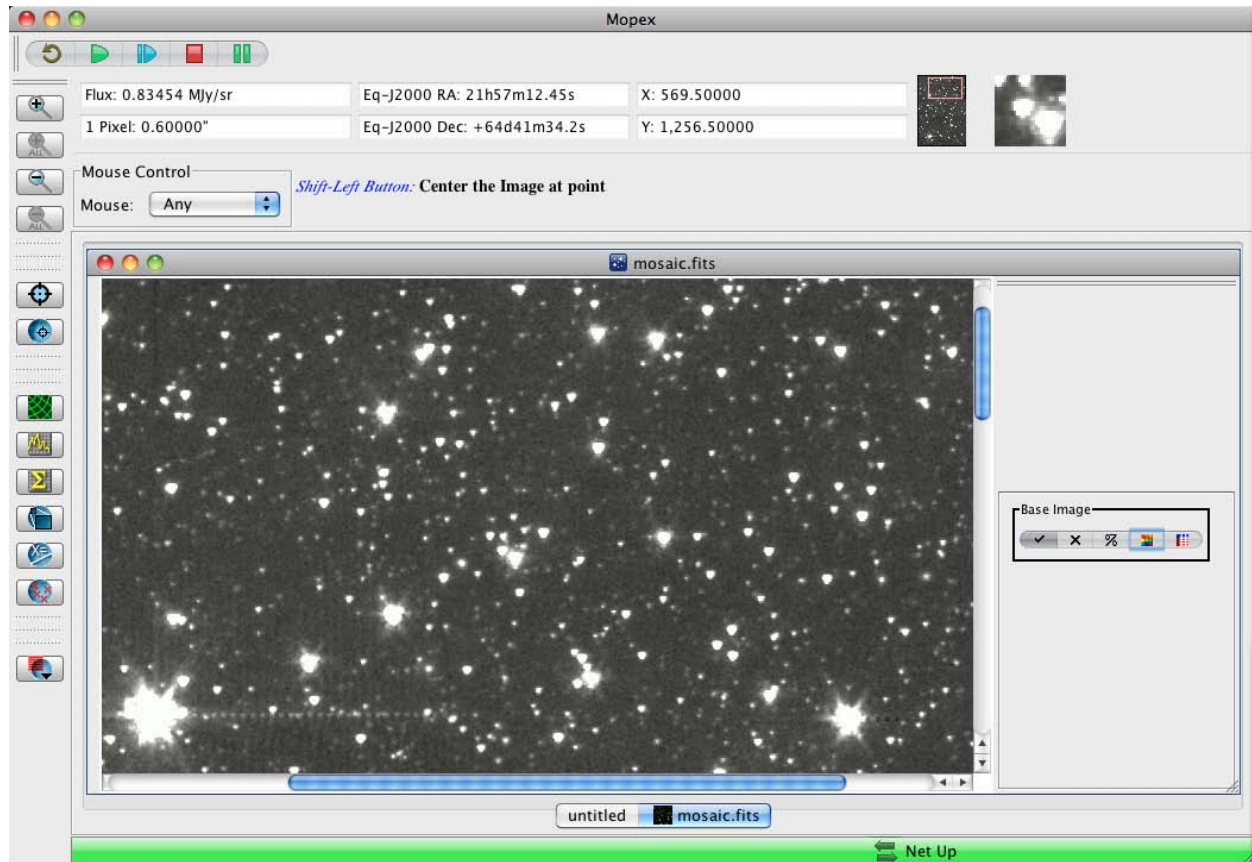
If desired, click on the Apex QA Settings. You will see it is expecting an Extract table: a list of positions and fluxes to subtract. Click "Cancel". The GUI will recognize that Apex is in the flow, and it will get the Extract table, FIF, and PRF from that run.

7.9 Run the Flow

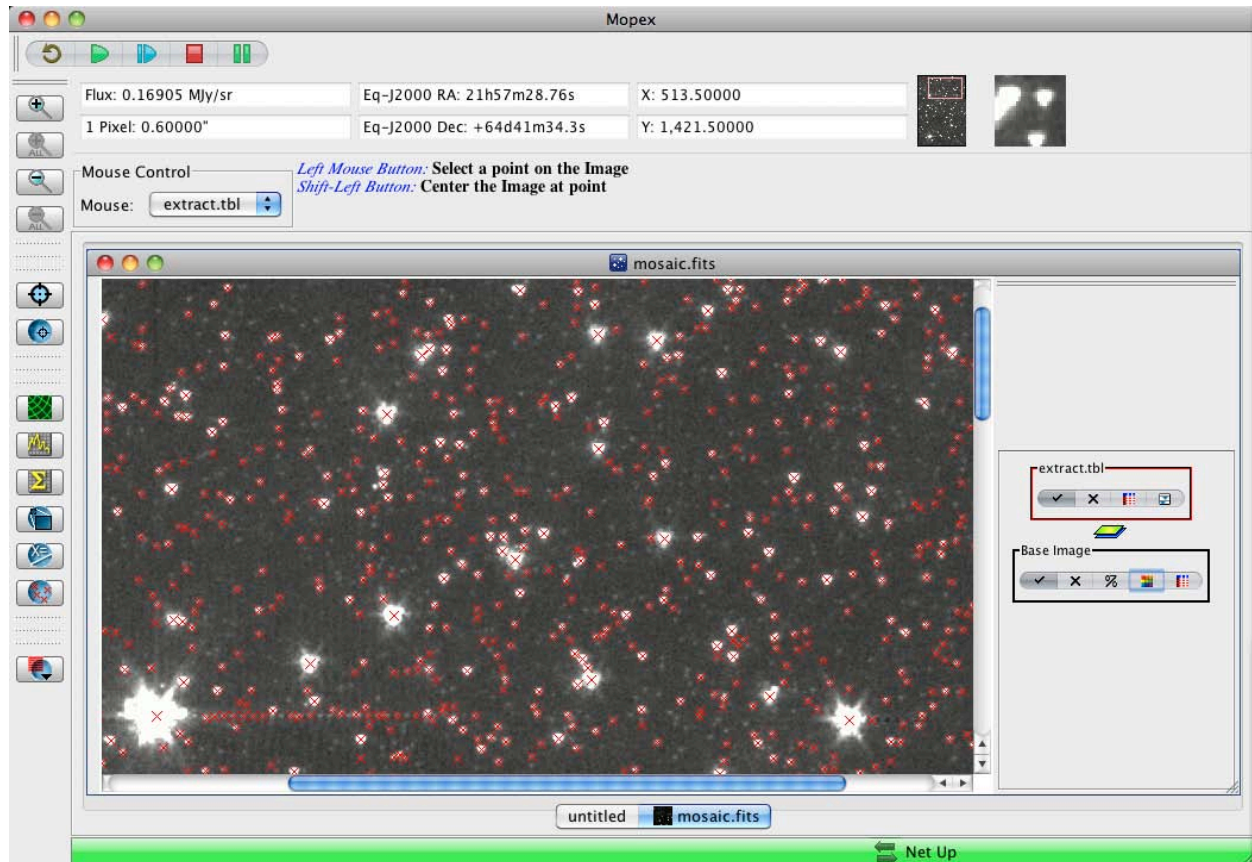
You are now ready to run the complete flow, end-to-end, from input (C)BCDs to the mosaic of image residuals! Click the large, green "arrow" button toward the top-left corner of the MOPEX GUI, and the flow will get underway. A query box will come up at the end asking whether you want to delete intermediate files. This is usually acceptable.



When the flow is completed, we can examine some of the output. Scroll back to the bottom of the Mosaic pipeline, to the Mosaic Combiner module, and click on "Mosaic image file". The mosaic you made (`<output>/Combine-mosaic/mosaic.fits`) is shown in the image window.

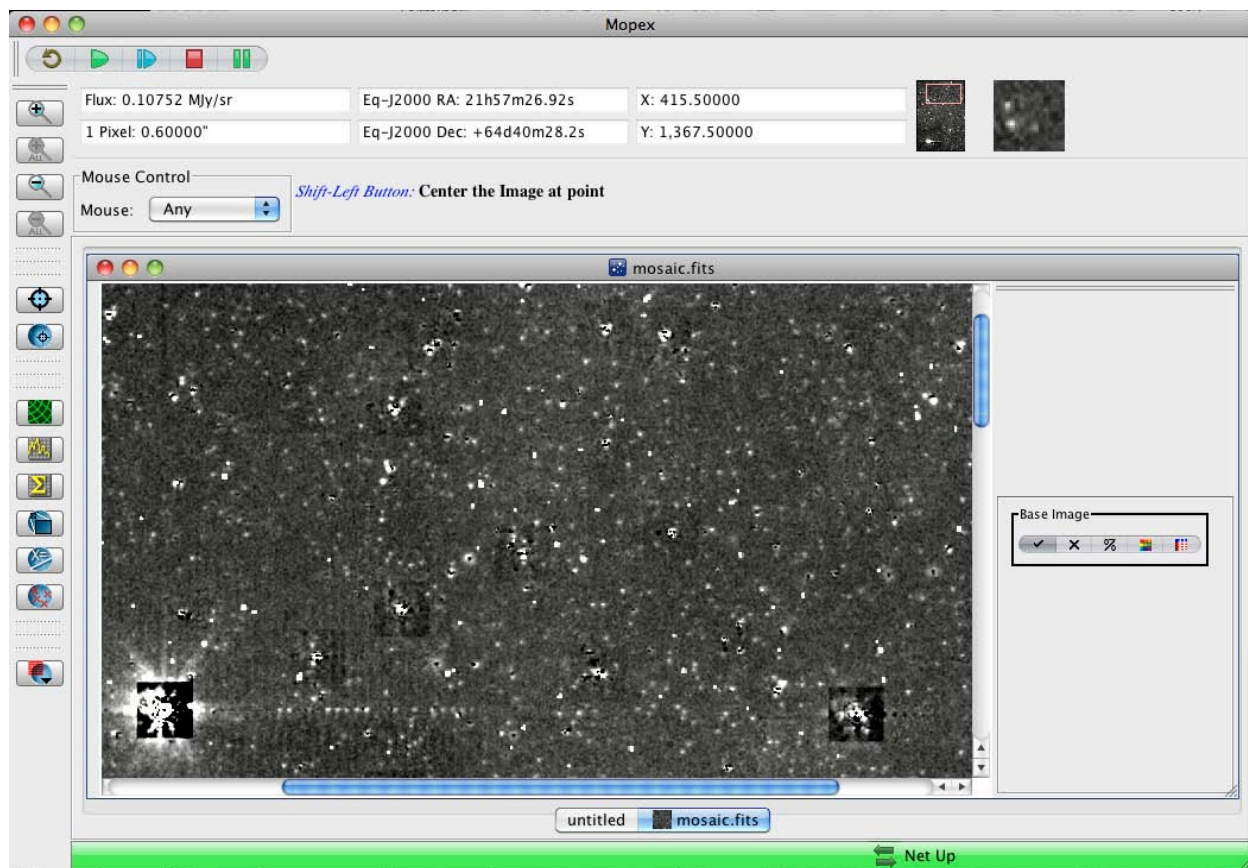


There are two ways to overlay the final extract table on the mosaic. The first is a general Overlay tool. Go up to Overlays in the top MOPEX menu bar, select "Catalog File" and choose `<output>/extract.tbl`. The sources will be overlaid on the mosaic. (An additional table file, `extract_raw.tbl`, contains the raw extractions before the Select module in APEX was run.) The second way is to scroll to the APEX flow, Select window, and click "Overlay Output Table on `mosaic.fits`."



As you can see, APEX with its default parameters did a good job of detecting point sources in the regions away from the bright objects. Around bright objects (or in complex backgrounds) it can sometimes have difficulty identifying true sources. So it is worth examining the results carefully in these regions. In the worst cases, you can input your own "Detect" list to the point-source photometry steps using the APEX User List module. Unfortunately, APEX also detected "sources" which are clearly artifacts created by the brighter sources (lower left of image above). These artifacts are the result of muxbleed, which is not completely corrected in the version of the IRAC pipeline used to process these images. An [IRAC artifact mitigation code](#), written in IDL, can be used to correct for muxbleed, after first correcting for source saturation (the routine "Iracworks," to perform saturation correction, can also be found at that site). For this recipe, we clearly did not run this artifact mitigation code. We recommend that you perform artifact mitigation on your IRAC (C)BCD data before running this flow and performing photometry.

We can check how well the PRFs fit the point sources. At the bottom of the APEX QA flow in the Mosaic Combine window, select "Mosaic Residual Images - Image File". This will display the residual mosaic (`<output>/Residual/mosaic/Combine/mosaic.fits`).



Finally, the fluxes in the *extract.tbl* file require two types of small corrections for the highest accuracy. The point-source fitted fluxes are in the column "flux" and the best uncertainty estimate is given as a signal-to-noise ratio in the column "SNR". Because of differences in normalization between IRAC data and APEX, a small correction needs to be applied to "flux". These are given in Appendix C.3, Table C.1 of the IRAC Instrument Handbook. For example, for IRAC1 divide "flux" by 1.021. For bright, isolated point sources, this correction should give photometry whose median is within about 2 % of standard aperture photometry.

The second correction is the array-position dependent color correction discussed above, and applies to both point-source fitting and aperture photometry (e.g. columns "aperture2" and "ap_unc2"). Use the correction mosaic you made in `<output>/Combine-mosaic/mosaic_arraycorr.fits` to correct photometry. (There is no MOPEX GUI tool for this at present.) Multiply final point-source fitted or aperture fluxes by the correction factor at the source position for blue (star-like) sources. The correction factor goes to 1 for red sources, where red is like the zodiacal emission. Corrections for intermediate colors can be found by interpolation.

You should now generally be able to apply this recipe to other IRAC channels and other science cases with relatively uniform backgrounds.

Recipe 8. Bandmerge: A How-To Example

This recipe provides a step-by-step guide on how to use the Bandmerge software.

8.1 Requirements

Bandmerge was designed to work on tables produced from Spitzer data by the SSC's MOPEX/APEX software which does photometry on images. The software can be downloaded from the SSC web site (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/bandmerge/>).

This recipe uses a sample dataset included in the Bandmerge package.

8.2 What does Bandmerge do?

Bandmerge merges source photometry lists from different wavelengths by matching source positions. It can read in source list tables for two to seven different wavelength bands and produce a merged list.

Bandmerge was designed primarily to work on photometry tables produced from Spitzer IRAC and MIPS data by the SSC's MOPEX/APEX software. Strictly speaking, APEX is not required, but Bandmerge does expect certain keywords and columns, and when it recognizes a Spitzer band, certain parameters are "hard-wired". For instance, Bandmerge will label Spitzer bands with an Outband number based on Spitzer wavelength order: IRAC (3.6, 4.5, 5.8, 8.0 microns) = 1 - 4. MIPS (24, 70, 160 microns) = 5 - 7. Bandmerge has some limited ability to work on other kinds of data. But this demo assumes you are working on Spitzer data, either IRAC or MIPS, or both. Sample APEX tables are provided.

There are several features that make Bandmerge more sophisticated than a simple position-matcher. One important one is that when performing positional matching, Bandmerge takes into account positional uncertainties, and it will estimate any systematic offsets between two different bands, correct this offset, and do more accurate source-matching iteratively.

For a more general, though less sophisticated, "closest-match" position-matching tool, see the included file *README_mgsa*.

Bandmerge is just a "best" position-matcher. It should work for uncrowded fields. Its weakness is close multiples. It will try to make unique one-to-one matches. A common failure might be a close pair at the shortest wavelength, a smeared source at the next longest, and a large blob at the longest. Bandmerge might assign the smeared source to one of the pair, the large blob to the other. Further analysis of close multiples is required of the user, taking fluxes into account.

Bandmerge consists of a Perl wrapper script, `bandmerge.pl`, which is called from the command line. It in turn calls C and Fortran binaries. The Bandmerge Perl script is most easily controlled by a namelist, e.g. `bandmerge.nl`, which needs to be in the `./cdf` sub-directory of the working directory.

Bandmerge will typically read in source positions in sky coordinates (RA, Dec) and project them onto a common fiducial image plane with corresponding (x,y) positions. The bandmerging is performed in the fiducial plane in (x,y), and the merged catalog with (x,y) positions is converted back to sky coordinates. The fiducial plane can be specified by the user with an `FIF.tbl`, like those generated by MOPEX, or Bandmerge can compute an `FIF.tbl` if the user provides a reference FITS image.

A required input file is called the band-pair registration uncertainty file. As mentioned above, Bandmerge will try to check for offsets between bands iteratively. It needs to start with a file, e.g. `bpru.tbl`, with some nominal initial guesses for the offset uncertainties. It should be in `cdf/`.

In the namelist file, you set how many iterations of this (bandmerge) - (get offsets) - (bandmerge) step you want to run. For each iteration, a separate namelist, `bmg1.nl`, `bmg2.nl`, etc., will control Bandmerge parameters. These namelists are in `cdf/`.

8.3 Setting up Bandmerge software and the example dataset

Within the downloaded Bandmerge package are both the software and a sample dataset.

8.3.1 Set up the environment for Bandmerge

Before starting Bandmerge, we need to set some environment variables. In the bandmerge directory, edit the first line in `bandmerge.csh` to your bandmerge directory; see the line highlighted in bold type.

```
setenv MOPEX_INSTALLATION /yourpath/bandmerge
setenv SIRTF_JAVA ${MOPEX_INSTALLATION}/java
setenv SIRTF_BIN ${MOPEX_INSTALLATION}/bins
setenv WRAPPER_UTILS ${MOPEX_INSTALLATION}/bins
setenv WRAPDIR ${MOPEX_INSTALLATION}/bins
setenv SIRTF_ANC ${MOPEX_INSTALLATION}/source
set path = ( ${MOPEX_INSTALLATION}/bins $path )
if ($?LD_LIBRARY_PATH) then
    setenv LD_LIBRARY_PATH ${MOPEX_INSTALLATION}/libs:${LD_LIBRARY_PATH}
else
    setenv LD_LIBRARY_PATH ${MOPEX_INSTALLATION}/libs
endif
if ($?PERL_PATH) then
else
    setenv PERL_PATH `which perl | sed "s/\(.*\)perl/\1/" `
endif
```

```
setenv SIRTF_QA cdf
setenv SIRTF_CDF cdf
setenv SIRTF_CAL cal
```

Then in a Cshell type: `source bandmerge.csh`.

NOTE: The *bandmerge.csh* file "borrows" some of the same environment variables used by MOPEX. If you also use MOPEX, it's best to start a new shell to run Bandmerge and exit it when done.

One can run the Perl script from the bandmerge directory, but it is often clearer to move to a separate "working" directory. For our example this is *working_demo/*.

For the demo: `cd ./working_demo`.

Note the sub-directories *data/* and *cdf/*. The data can be anywhere, provided a path is given. The namelists, *bandmerge_demo.nl* and *bmg1.nl* and *bmg2.nl*, are in *cdf/*. The initial band-pair registration uncertainty file, *bpru.tbl*, is there as well. Log files for the bandmerge and getoff modules will also go into *cdf/*. These log files are very useful for identifying the sources of failures. The final merged source list, *newtbl.tbl*, will be in the user-specified Output directory.

8.3.2 Dataset for this example

In *data/* are sections of images in IRAC channels 1, 3, and MIPS 24 microns from the S-COSMOS Legacy Science Program (PI: D. Sanders). The dataset includes images, coverage and uncertainty maps, with names like *irac1.fits*, *irac1_cov.fits*, and *irac1_unc.fits*. We will choose *irac1.fits* to be the reference image for the fiducial plane. There are also APEX extraction tables, e.g. *irac1_extract.tbl*.

8.4 How to use Bandmerge

8.4.1 The input source files

Bandmerge required input file format:

Bandmerge can only take input data in IPAC table format with the first 8 required columns shown below. Additional columns are allowed. The Perl script does a check of the headers. Specifically, it will count the actual number of sources in the input file, and internally add or update the keyword **Total_PS_Number** if it is not consistent. Similarly, keywords **CDEL1,2** and **NAXIS1,2** are taken from the reference FITS image header or the FIF table. The Bandmerge namelist also has a block of parameters which maps out the instruments and channel numbers, and keywords **INSTRUME** and **CHNLNUM** are added internally as needed.

Aperture fluxes are usually part of an APEX run and one generally wants them in the final bandmerged

table. So Bandmerge will carry along the number of apertures given by **N_Apertures**. Both the flux value for aperture n "aperture(n)" and the bad area in pixels "bad_pix(n)" will be carried along.

An optional input column is flux signal-to-noise ratio, SNR, which Bandmerge can use to bin sources into different groups while computing statistics.

```

\int Total_PS_Number = 714
\float CDELTA1 = -1.6667e-04
\float CDELTA2 = 1.6667e-04
\int NAXIS1 = 501
\int NAXIS2 = 501
\char INSTRUME = IRAC
\int CHNLNUM = 1
\int N_Apertures = 1
|srcid |flux   |delta_flux |RA      |delta_RA  |Dec      |delta_Dec |delta_RAD |aperture1|bad_pix1|
|l      |r       |r          |d       |r         |d        |r         |r         |r         |r        |
1      |25.233  |3.490     |54.229270|0.000006  |-29.835540|0.000009  |-0.000002|12.610   |0.0
2      |6.376   |0.969     |54.215940|0.000011  |-29.833250|0.000016  |-0.000007|4.457    |0.0

```

APEX tips for Bandmerging (not required for demo):

The most direct way to generate input files with the appropriate format is to use the SSC's photometry extraction software APEX. For descriptions on how to use APEX, refer to the MOPEX documentation (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/mopex/>). As described above, Bandmerge requires certain columns of data. With APEX, make sure the "select" module is allowing these columns through to the final output *extract.tbl*. (If using the GUI on Mac computers, use Apple-click to toggle selected columns; for Unix systems, use Control-left mouse.)

8.4.2 Prepare to run Bandmerge

Defining the fiducial image plane

In this test case, we do not have an *FIF.tbl* file, so we will have Bandmerge generate one using the IRAC channel 1 image as the master reference. To do this, comment out the FIF_FILE_NAME and specify in the namelist:

```

REFERENCE_FITS_FILENAME = data/irac1.fits

#FIF_FILE_NAME = myfif.tbl

```

This will produce an output *FIF.tbl* file in the working directory like the following:

```

\char comment=fif.tbl created by make_fif in bandmerge script
\real CRVAL1=150.092822
\real CRVAL2=2.1826860
\real CRPIX1=-3569.926508
\real CRPIX2=-3450.384412

```



```

\int NAXIS1=501
\int NAXIS2=501
\real CDELTA1=-0.000166667
\real CDELTA2=0.000166667
\real CROTA2=0
\char CTYPE1=RA---TAN
\char CTYPE2=DEC--TAN
\char PROJTYPE=TAN
\real EXTENT_X=0.083500167
\real EXTENT_Y=0.083500167

```

Band-pair registration uncertainty file:

Bandmerge requires an initial band-pair registration uncertainty file, *bpru.tbl*, in *cdf/*. Here Columns 1 and 2 specify the pair of bands, Columns 3, 4 and 5 are the positional shift uncertainties between two bands in X and Y and a cross term shift (XY). There is a default file typical for the Spitzer bands provided as a part of the bandmerge package. Using the default file provided should work for most users. This file can contain the initial offset values for each pair of 7 bands, even though you might be using fewer bands.

```

\char comment = Spitzer band-pair registration uncertainties
\ Generated by getoff vsn 1.91 A60816 on Thu Jan 11 12:16:05 2007
|seed_index |cand_index |X_sigma |Y_sigma |XY_csd |
|int |int |real |real |real |
1 2 0.15 0.15 0.0
1 3 0.15 0.15 0.0
1 4 0.15 0.15 0.0
1 5 0.15 0.15 0.0
1 6 2.00 2.00 0.0
1 7 2.00 2.00 0.0
2 3 0.15 0.15 0.0
2 4 0.15 0.15 0.0
2 5 0.15 0.15 0.0
2 6 2.00 2.00 0.0
2 7 2.00 2.00 0.0
3 4 0.15 0.15 0.0
3 5 0.15 0.15 0.0
3 6 2.00 2.00 0.0
3 7 2.00 2.00 0.0
4 5 0.15 0.15 0.0
4 6 2.00 2.00 0.0
4 7 2.00 2.00 0.0
5 6 2.00 2.00 0.0
5 7 2.00 2.00 0.0
6 7 2.00 2.00 0.0

```

The uncertainty values will be recalculated by Bandmerge on each iteration and new *bpru.tbl* files written.

8.4.3 Set Bandmerge Parameters:

Bandmerge is controlled by a set of namelists, which are stored in the *cdf/* directory. The first, e.g. *bandmerge_demo.nl*, controls the run. If you specify in *bandmerge_demo.nl* that you want to run the bandmerge - getoffset - bandmerge step iteratively 2 times, you will need another two namelists (with fixed names), *bmg1.nl* and *bmg2.nl*, also in *cdf/*.

Most of the namelist parameters should give acceptable results as is. Arrows describe some important parameters. For other data, edit the PointSourceList_FileNames, REFERENCE_FITS_FILENAME, and Instrument_Channel ID's.

8.4.4 *bandmerge.nl*

This namelist is required to run *bandmerge.pl*. It looks like the following:

```
convert_sky_to_cartesian = 1    ----->  this switch converts (RA DEC)
into (X Y) positions
run_bandmerge = 1              ----->  this turns on/off the bandmerge
module.
convert_cartesian_to_sky = 1    ----->  this switch converts (X Y) to
(RA DEC) in the final merged output.

#input files, PointSourceList_FileName has to be sequential
#
PointSourceList_FileName1 = data/irac1_extract.tbl  ---> you can use
the absolute path if the data are not in the current path
PointSourceList_FileName2 = data/irac3_extract.tbl
PointSourceList_FileName3 = data/irac4_extract.tbl

# input to define the fiducial plan
# bandmerge perl script will look for either one of these two input
parameters,
# if both parameters are not found, bandmerge will stop.
REFERENCE_FITS_FILENAME = data/irac1.fits          ---> you can
use the absolute path if the data are not in the current path
#FIF_FILE_NAME = fif.tbl                          ---> this has been commented
out since we are using an image as reference

# map out the channels with Spitzer ID's
Instrument_Channel1 = IRAC_1      ----->  IRAC_1 = 3.6, IRAC_2 = 4.5,
IRAC_3 = 5.8, IRAC_4 = 8.0 (microns)
Instrument_Channel2 = IRAC_3
```

```

Instrument_Channel3 = IRAC_4      -----> MIPS_1 = 24, MIPS_2 = 70,
MIPS_3 = 160 (microns)

Input_BPRU_Filename = bpru.tbl   -----> this is the initial band pair
registration uncertainty file.
                                     -----> It can contain more bands
than actually used.

#needed for iterations. For no iteration, set Number_of_iterations = 0
Number_of_iterations = 2         ----> number of iteration of
bandmerge-getoffset-bandmerge. Cannot be zero.
#correction type used in iterations
clean_type = 3                  ----> value of 1 means that the derived
offsets are to be applied to
                                     -----> the source positions, value of
2 allows updates in (X,Y)
                                     -----> positional variances, value of
3 allows updates in both (X,Y)
                                     -----> positions and uncertainties.

#default is 's2c'
s2c_prefix = 'cnv'              ----> the prefix to the files
after (RA,Dec) have been converted to (X,Y).
OUTPUT_DIR = output_bandmerge_demo ----> give absolute path if not in
the current path
#file of merged sources
OUTPUT_FILE_NAME = 'newtbl.tbl'  ----> Note the final output file is
called newtbl.tbl, the intermediate output file is called
iter1_newtbl.tbl and iter2_newtbl.tbl. This file will be in the output
directory.

# the following parameter block is for bandmerge module.
# Most parameters can be set as the default values as shown here
# Chi_Sq_Max can be tuned to perform more stringent matches.
#
&BANDMERGEIN
  Comment = 'Generic namelist file for bandmerge - default values.',
  Output_STAT_Filename = 'statfile.tbl',
  Output_DUMP_Filename_Base = 'bm_dump',
  Output_SPCMB_Filename = 'spcmbfile.bm',
  Project = 1,
  ChiSq_Denom_Min = 1.0e-06,
  Chi_Sq_Max = 24.0,             -----> Maximum allowed chi-square
value used to specify the match.
  Pseudo_Chi_Sq_Max = 24.0,

```

```

LR_Scale_Fact = 2.5,
Epsilon = -1.0e+39,
SigMin = 0.03,
Pseudo_Pos_SF = 1.0,
Pseudo_Phot_SF = 0.0,
X_Window = 20.0,
Y_Window = 20.0,
Status_Check_Flag = 0,
Status_Threshold = 0,
Dump = 0,
Trace_PS1 = 0,
Trace_PS2 = 0,
Trace_PS3 = 0,
Trace_PS4 = 0,
Trace_PS5 = 0,
Trace_PS6 = 0,
Trace_PS7 = 0,
SNR_Threshold1 = 10.0,
SNR_Threshold2 = 10.0,
SNR_Threshold3 = 10.0,
SNR_Threshold4 = 10.0,
SNR_Threshold5 = 10.0,
SNR_Threshold6 = 10.0,
SNR_Threshold7 = 10.0,
#Input_Table_Column1 = 'SNR',          -----> To bin by SNR, put SNR
column in input files and uncomment
SNR_Bin1 = 20.0,
SNR_Bin2 = 50.0,
SNR_Bin3 = 100.0,
SNR_Bin4 = 200.0,
SNR_Bin5 = 500.0,
SNR_Bin6 = 1000.0,
SNR_Bin7 = 2000.0,
SNR_Bin8 = 5000.0,
SNR_Bin9 = 10000.0,
NL_print = 0,
&END

```

8.4.5 *bmg1.nl* and *bmg2.nl* -- two additional namelists for subsequent iterations 1 and 2.

During iterations to determine band pair systematic offsets, the bandmerge script is controlled by namelists with fixed names, *bmg1.nl* and *bmg2.nl*, and so on. In our example the only difference between these files is that the `Chi_Sq_Max` parameter is set to be smaller for the first pass of bandmerge than for the second pass. The reason is that in order to get reliable estimates of the systematic offsets between pair

of bands, it is better to use only very good matches, thus a smaller chi square value. After any offsets between the band pairs have been computed and applied, one can allow larger chi square values to get more matches. The two example files are shown below:

8.4.6 *bmg1.nl*

```
&BANDMERGEIN
Project                = 1,
ChiSq_Denom_Min       = 1.0e-06,
Chi_Sq_Max            = 12.0,
Pseudo_Chi_Sq_Max    = 12.0,
Epsilon               = -1.0e+39,
SigMin                = 0.03,
Pseudo_Pos_SF        = 1.0,
Pseudo_Phot_SF       = 0.0,
LR_Scale_Fact        = 2.5,
X_Window              = 20.0,
Y_Window              = 20.0,
Status_Check_Flag    = 0,
Status_Threshold     = 0,
SNR_Threshold1       = 10.0,
SNR_Threshold2       = 10.0,
SNR_Threshold3       = 10.0,
SNR_Threshold4       = 10.0,
SNR_Threshold5       = 10.0,
SNR_Threshold6       = 10.0,
SNR_Threshold7       = 10.0,
SNR_Bin1              = 20.0,
SNR_Bin2              = 50.0,
SNR_Bin3              = 100.0,
SNR_Bin4              = 200.0,
SNR_Bin5              = 500.0,
SNR_Bin6              = 1000.0,
SNR_Bin7              = 2000.0,
SNR_Bin8              = 5000.0,
SNR_Bin9              = 10000.0,
Warning_Max          = 10,
NL_print              = 1,
&END
```

8.4.7 *bmg2.nl*

```
&BANDMERGEIN
Project                = 1,
ChiSq_Denom_Min       = 1.0e-06,
```

```

Chi_Sq_Max           = 24.0,
Pseudo_Chi_Sq_Max   = 24.0,
Epsilon              = -1.0e+39,
SigMin               = 0.03,
Pseudo_Pos_SF       = 1.0,
Pseudo_Phot_SF      = 0.0,
LR_Scale_Fact       = 2.5,
X_Window             = 20.0,
Y_Window             = 20.0,
Status_Check_Flag   = 0,
Status_Threshold    = 0,
SNR_Threshold1      = 10.0,
SNR_Threshold2      = 10.0,
SNR_Threshold3      = 10.0,
SNR_Threshold4      = 10.0,
SNR_Threshold5      = 10.0,
SNR_Threshold6      = 10.0,
SNR_Threshold7      = 10.0,
SNR_Bin1            = 20.0,
SNR_Bin2            = 50.0,
SNR_Bin3            = 100.0,
SNR_Bin4            = 200.0,
SNR_Bin5            = 500.0,
SNR_Bin6            = 1000.0,
SNR_Bin7            = 2000.0,
SNR_Bin8            = 5000.0,
SNR_Bin9            = 10000.0,
Warning_Max         = 10,
NL_print            = 1,
&END

```

8.5 Output from Bandmerge

In our example, we have *irac1_extract.tbl*, *irac3_extract.tbl*, *irac4_extract.tbl* as the input files to Bandmerge, and they are in *working_demo/data/*. The namelist for bandmerge is */cdf/bandmerge_demo.nl*. Now we run Bandmerge, by typing (in *working_demo/*):

```
prompt% bandmerge.pl -n bandmerge_demo.nl
```

This will give screen output like:

```

prompt% bandmerge.pl -n bandmerge_demo.nl
=====

```

Wrapper-script bandmerge.pl, Version 1.5

```
path /stage/ssc-pipe/work/set11/postbcd/irk/mopex/tst_bins
input ref fits = iraclsci.fits
  naxes = 3246 4103
  crvals = 150.092822 2.182686
  ctypes  RA---TAN DEC--TAN
cdelts = -0.000166667 0.000166667 crota2 = 0.
proj = TAN extents = 0.541001082 0.683834701
Executables pathname = /stage/ssc-
pipe/work/set11/postbcd/irk/mopex/tst_bins/
Ancillary-data pathname = /stage/ssc-
pipe/work/set11/postbcd/irk/mopex/source/
-r file name = fif.tbl
-i file name = iraclsci_extract.tbl
-o file name = Output_1/cnv_iraclsci_extract.tbl

/stage/ssc-pipe/work/set11/postbcd/irk/mopex/tst_bins/s2c_trans -r
fif.tbl -i iraclsci_extract.tbl -o Output_1/cnv_iraclsci_extract.tbl
```

```
Pipeline Module S2C_TRANS Version 1.8
Processing time Tue Jul  8 09:30:17 2008
```

```
.
.
.
```

```
Pipeline Module C2S_TRANS Version 2.0
Processing time Tue Jul  8 09:31:26 2008
```

```
c2s_trans_parse_namelist: Information only: Namelist not specified.
Time = 8.420 seconds
exit code 0
```

```
Perl system() return code = 0
c2s_trans terminated normally.
```

```
System Exit Code (                c2s_trans): 0
```

```
Wrapper-script bandmerge.pl terminated normally.
```

Note: On Mac OS X and Linux systems, ignore the messages "m: command not found" near the end. These are just some timing scripts that are not vital.

After executing the Bandmerge Perl script, let us look at the output. All of the log files are stored in the *cdf/* directory. Look at these to help diagnose errors. The science outputs are stored in the Output directory which was specified in the namelist, *output_bandmerge_demo/*. These outputs are:

8.5.1 Intermediate output files:

In the example the intermediate source lists have a prefix given in the namelist, *cnv_**, for example, *cnv_irac1_extract.tbl*.

Bandmerge derives any systematic offsets between pair of bands, and corrects the input positions and positional uncertainties. The corrected files are called, for example, *iter1_cnv_irac1_extract.tbl*, after the first iteration of bandmerge and getoff, and *iter2_** for the second iteration.

8.5.2 Merged source lists:

The name of the merged source photometry files is determined by the user's input in the namelist. In this example, we specified *newtbl.tbl* in the namelist. After the first iteration, the merged source photometry file is called *iter1_newtbl.tbl*. It's the final merged source list that is called *newtbl.tbl*.

The final merged source list has a format like the following. Note how Bandmerge assigns the Outband number based on Spitzer wavelength order (IRAC 1-4, MIPS 5-7):

```

\BANDMERGE Ver 1.39 A50707 run on 07/08/08 at 10:09:58
\INT Nbands = 3
\INT Total_Merged_Number = 68619
\CHAR Infile-1 = /ssc/pbcd/lyan/test/Output_1/iter2_cnv_irac1test.tbl
\CHAR Infile-2 = /ssc/pbcd/lyan/test/Output_1/iter2_cnv_irac3sci_extract.tbl
\CHAR Infile-3 = /ssc/pbcd/lyan/test/Output_1/iter2_cnv_mips24sci_extract.tbl
\INT N_Apertures-1 = 0
\INT N_Apertures-2 = 0
\INT N_Apertures-3 = 0
\CHAR Sort_& Search_Direction = Y
\CHAR Outband-1 = Infile-1
\CHAR Outband-3 = Infile-2
\CHAR Outband-5 = Infile-3
\INT InBlank = -999
\REAL RealBlank = -9.9e+09
\CHAR CharBlank = null
|JDNnum |X |Delta_X |Y |Delta_Y |Delta_XY |CnfFlg |srcid1|flux1 |deltflux1 |srcid3|flux3 |deltflux3 |srcid5|flux5 |deltflux5 |RA |Delta_RA |Dec |Delta_Dec |Delta_RAD |
|int |real |real |real |real |real |char |int |real |real |int |real |real |int |real |double |double |double |double |double |
1 |1499.437 |3.779e-01 |1.187 |3.889e-01 |8.333e-02 |000000 |46 |7.10500e+00 |6.64e-02 |0 | -9.9e+09 | -9.9e+09 |0 | -9.9e+09 | -9.9e+09 |1.50016205e+02 |6.2983e-05 |2.18794596e+00 |6.4817e-05 |1.38883608e-05
274 |2644.048 |3.693e-01 |18.917 |5.546e-02 |1.749e-02 |101000 |229 |1.07800e+02 |2.23e-01 |467 |4.49600e+01 |4.73e-01 |0 | -9.9e+09 | -9.9e+09 |1.49825299e+02 |6.1550e-05 |2.19087906e+00 |9.2434e-06 |2.91500575e-06
1883 |1691.435 |3.589e-01 |113.855 |1.543e-01 |4.052e-02 |111111 |1866 |1.58800e+01 |7.49e-02 |3577 |1.12000e+01 |3.67e-01 |86 |7.12000e+01 |4.38e+00 |1.49984181e+02 |5.9817e-05 |2.20672199e+00 |2.5717e-05 |6.75334704e-06

```

For a larger version of this image, please see

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisstools/cookbook/files/Recipe5.gif>

How to check merged entries in the final merged list:

The 7th column, CnfFlag, indicates the source matching results among the three bands. The flag has a width of Nbands*(Nbands-1). In the above example, Source 1 has CnfFlag=000000. The first 2 digits (00) mean the lowest-wavelength band IRAC-3.6um (Outband 1) had no matches in IRAC-5.8um (Outband 3) and MIPS 24um (Outband 5). This is why srcid3 and srcid5 are zero. For source 274, CnfFlag = 101000, where the first two digits (10) mean IRAC-3.6um got a match in IRAC-5.8um, but not in MIPS 24um, and the second two digits (10) mean that IRAC-5.8um got a match in IRAC-3.6um, but not in MIPS 24um, and the third two digits (00) mean no MIPS 24um detection.

Extrapolating the above definition to 4 bands, if a source has CnfFlag=1111111111, the first set of (111) means that the lowest-wavelength channel has a unique match with the other 3 wavelengths, and the

second set of (111) means that the second lowest-wavelength channel has a match with the first band plus the two longest bands, etc.

To select sources with a given match, the user can either parse the CnfFlag (called CF if only 2 bands), or read in fluxes and drop the ones that are $-9.9e+09$. Close multiples are difficult for Bandmerge. The user may need to check sources for nearby companions and examine the matches, perhaps taking fluxes into account.

8.5.3 Diagnostic files:

Bandmerge also outputs two statistical files in the Output dir. The first one is called *spcmbfile.bm* (the name can be adjusted in the namelist). It gives the number of matched sources in each band combination:

```
Sp. Comb.   Count
=====
    001     50471
    010      3315
    011     11186
    100       156
    101       833
    110       359
    111     2299
=====
Total:      68619
```

The digits of the first column correspond to bands in descending order, with 0 for no match and 1 for a match. Here, a total of 2299 sources got matched in all three bands, and 50471 sources had only IRAC chan1 detections

The second diagnostic file is *statfile.tbl*, whose name can also be adjusted in the namelist. This file gives some statistics about each two-band matched set in the x,y plane. The second column indicates that no binning by SNR was done - these are statistics for the Total set.

```
\ BANDMERGE Vsn 1.39 A50707 run on 07/08/08 at 10:09:58
\CHAR BMG_Output_file = /ssc/pbcd/lyan/test/Output_1/newtbl.tbl
|B-P |SNR_bin |AveDX |AveDY |StdDevX |StdDevY |ChSqX |ChSqY |ChSqXY |NSum |
|c |char |real |real |real |real |real |real |real |int |
1-3 Total 0.05022 -0.02733 3.54535 1.27333 0.03611 1.09304 1.12867 4728
1-5 Total -0.01574 -0.00661 1.57775 1.77159 0.95505 0.58009 1.54111 760
3-5 Total -0.08312 -0.02329 2.08067 1.63772 0.10594 0.64384 0.75379 585
```

Recipe 9. Spitzer Synthetic Photometry

If you have a template source spectrum, you may wish to predict the calibrated flux densities that would be reported in the various Spitzer bands (e.g. IRAC 3.6, 4.5, 5.8, 8.0 microns; IRS Peakup Imaging 16, 22 microns; MIPS 24, 70, 160 microns). This can be useful, for instance, in deriving photometric redshifts. After introducing the relevant basic concepts, this section provides step-by-step instructions for determining synthetic, calibrated IRAC, IRS, and MIPS photometry from such a template.

9.1 Requirements

In this section we have collected information which will be useful to those who wish to compute synthetic Spitzer photometry. We also provide IDL code (only tested on IDL version 6.3) which can calculate synthetic fluxes for smooth spectra (generally not pure emission lines). Examples using the code are provided. The IDL code is available on the SSC web site here:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/spitzer_synthphot_28Oct2008.tar.gz.

Unpack the zipped tar file in a directory that is included in your IDL path:

```
gunzip spitzer_synthphot_28Oct2008.tar.gz
tar xvf spitzer_synthphot_28Oct2008.tar
```

Among the files you just unpacked are the filter response files (ending in *.txt* and *.dat*). These are assumed to be in your working directory in the following examples.

Note that this IDL code provides good agreement with the values tabulated in the IRAC, MIPS and IRS Instrument Handbooks. The results are consistent to within half a percent. The fact that the values are not exactly the same is likely due to differences in the interpolation and integration methods used. The differences in color correction are comparable to the errors in the measurement of the spectral response curves and will be less than the systematic error due to real differences between the assumed spectral shape and the observed object.

Note that this IDL code computes color corrections for the IRS peak-up filters as calibrated in pipeline versions S17 and greater. It is not to be used for data reduced with older versions of the pipeline.

9.2 The Basics

9.2.1 Useful Definitions

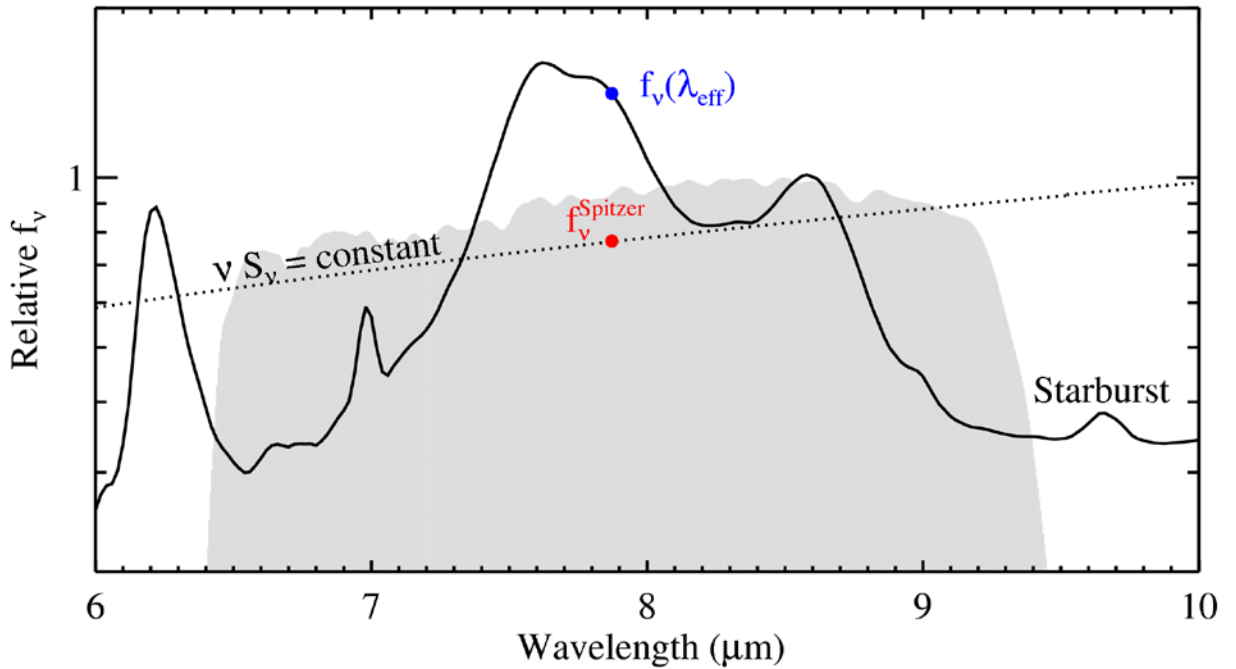
There are three quantities that are essential for understanding how to perform synthetic photometry for the Spitzer instruments on a given template spectrum.

1. λ_{eff} : This is the effective wavelength of the filter. These are defined and tabulated for each Spitzer instrument in the step-by-step guides below.
2. $f_{\nu}(\lambda_{\text{eff}})$: This is the monochromatic (measured at the effective wavelength of the filter) flux density of the template spectrum.
3. f_{ν}^{Spitzer} : This is the calibrated Spitzer flux density that would be reported (by e.g. MOPEX/APEX) if the template spectrum were observed by one of Spitzer's imagers. It corresponds to the monochromatic (measured at the effective wavelength of the filter) flux density of a source that
 - a. has a standard spectral shape (defined independently for each filter; see the step-by-step guides below); and
 - b. results in the same observed counts as the template source.

Thus, $f_{\nu}^{\text{Spitzer}} = f_{\nu}(\lambda_{\text{eff}})$ if the template spectrum has the standard spectral shape. However, significant deviations can occur if the template, or source, spectrum is very different than the standard, or reference, spectrum.

4. **K**: This is the color correction, which relates the above two quantities according to: . The color correction is computed in a different way for each of Spitzer's imagers (see the step-by-step guides below). It requires knowledge of 1) the source spectrum, 2) the reference spectrum, and 3) the filter response.

The plot below illustrates these concepts. For this example, we use the average starburst spectrum (http://www.strw.leidenuniv.nl/~brandl/SB_template.html) from Brandl et al. 2006 (<http://adsabs.harvard.edu/abs/2006ApJ...653.1129B>). We wish to determine the calibrated IRAC flux density that would be reported if this template source were observed in channel 4 (8 microns). The grey shaded region shows the arbitrarily-normalized response function for the IRAC channel 4 filter, in units of electrons per photon. The blue point shows $f_{\nu}(\lambda_{\text{eff}})$, which is simply the value of the template spectrum at the effective wavelength. However, since a starburst galaxy has a very different shape than the reference spectral shape for IRAC ($\nu S_{\nu} = \text{constant}$), f_{ν}^{Spitzer} is significantly different from $f_{\nu}(\lambda_{\text{eff}})$. To figure out how different, we must find the constant that will make the total counts from a $\nu S_{\nu} = \text{constant}$ spectrum the same as that from the starburst galaxy. The appropriately-normalized flat spectrum is shown as a dotted line in the figure below. To find f_{ν}^{Spitzer} , we simply evaluate this normalized flat spectrum at the effective wavelength. The result is shown as a red point. The ratio of these two quantities is just the color correction, which in this case is .



The IDL code used to create the above graph is available on the Spitzer web site here:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/spitzer_synthphot_example.pro.

9.2.2 Differences between IRAC, IRS, and MIPS photometry

The background information provided above applies to all Spitzer photometry from the IRAC, IRS, and MIPS instruments. However, the details are different for each instrument. The two main differences are:

1. The filter response curves are given in units of electrons per unit energy for MIPS and in units of electrons per photon for IRAC and IRS. Therefore, the equation used to calculate the color correction from the provided response function is different for MIPS than for IRAC and IRS. These differences are outlined in the step-by-step procedures provided below for each instrument.
2. The reference spectrum has the shape of a 10,000K blackbody for MIPS and that of a flat spectrum ($\nu S_\nu = \text{constant}$) for IRAC and IRS.

9.2.3 Converting from Spitzer photometry to other systems

In modern astronomical literature there are several reference systems in common use (SI, CGS, Jy, AB, Vega, etc). Each of these systems adopts a different reference spectrum. We have noted above that the IRAC and IRS systems adopt a reference spectrum described by $\nu S_\nu = \text{constant}$, while the MIPS system is based on a 10,000 Kelvin blackbody. In contrast, the AB and Jansky systems assume $\nu S_\nu = \text{constant}$, while the Vega system uses a spectrum of Vega, of which there are several different versions.

You may wish to put your Spitzer photometry on one of these other systems. You can use the basic concepts on this page to do so. However, be aware that the choice of a reference spectrum affects the

effective wavelength. When reporting photometry, the effective wavelength of the output system should be indicated. These will in general differ from the effective wavelengths presented on this page.

9.3 Procedure for computing synthetic IRAC photometry

9.3.1 *Determine your input spectrum.*

If you are going to calculate the correction using the IDL code, this should be in Janskys (i.e. energy flux per unit frequency) as a function of wavelength ($f_{\nu}(\lambda_e)$). The input spectrum should cover the width of the IRAC filter profile, so long as it has flux there. It should not be zero at the effective wavelength, and thus should generally not be a pure emission-line spectrum.

For example, you could use IDL to generate a power-law spectrum sampled at the same wavelengths as the IRAC channel 1 filter response function (you will need the IDL Astro Library:

<http://idlastro.gsfc.nasa.gov/>):

```
IDL>readcol, 'irac_tr1_2004-08-09.dat', filter_w, filter_t, format =
'(d, d)', /silent
IDL>alpha = 2. ; define the exponent of the power-law
IDL>c = 2.9979d8 * 1d6 ; speed of light in micron/s
IDL>wave = filter_w ;sample the power-law at the same wavelengths as
the filter
IDL>nu = c / wave
IDL>spec_fnu = nu^alpha / 1e28
IDL>plot, wave, spec_fnu, xtit = 'Wavelength (microns)', $
      ytit = 'Flux Density (Jy)', charsize = 1.5, /ylog, $
      yrange = [0.5, 1.], /ystyle ;plot the power-law spectrum
IDL>oplot, filter_w, filter_t / max(filter_t), linestyle = 1
;overplot the normalized response curve
```

9.3.2 *Estimate the flux density of the input spectrum at the effective wavelength.*

Given that the IRAC photometry is referenced to a spectrum with $\nu S_{\nu} = \text{constant}$, the effective wavelength is defined as:



This is equation 4.11 in the IRAC Instrument Handbook. For reference, the effective wavelengths of the IRAC filters are:

3.550 microns for channel 1

4.493 microns for channel 2
5.731 microns for channel 3
7.872 microns for channel 4

For the example, we could draw the effective wavelength on the plot:

```
IDL>lambda_eff = 3.550  
IDL>oplot, [lambda_eff, lambda_eff], 10.^(!y.crange), linestyle = 2
```

We can also estimate the flux density of the input spectrum at λ_{eff} :

```
IDL>linterp, wave, spec_fnu, lambda_eff, fnu_lambda_eff  
IDL>plotsym, 0, 1, /fill  
IDL>plots, lambda_eff, fnu_lambda_eff, psym=8
```

9.3.3 Determine the color correction, which for IRAC is defined by the following equation.



Here, $S_v(\lambda_{\text{eff}})$ is the reference spectrum evaluated at the effective wavelength; $f_v(\lambda_{\text{eff}})$ is the template spectrum evaluated at the effective wavelength; and $R_\gamma(v)$ is the filter response function as a function of frequency in units of electrons per photon.

Note that this expression is equivalent to equation 4.8 in the IRAC Instrument Handbook.

You can evaluate the expression above on your own by first ensuring that the response curve and the template spectrum (and reference spectrum) are on the same grid. You can choose an arbitrary normalization for the flat IRAC reference spectrum (constant). However, two more convenient options are available:

- A. If your input spectrum is tabulated in Section 4.4 of the IRAC Instrument Handbook, you can use the color correction from the tables. Then the expected Spitzer flux density is just given by:

$$f_v^{\text{Spitzer}} = f_v(\lambda_{\text{eff}}) \times K$$

- B. You can calculate the color correction using the IDL code as follows.
 - i. Obtain the IRAC response curves, . Current versions have been provided in the download directory with names *irac*.dat*. For updates check:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irac/calibrationfiles/spectralresponse/>.

- ii. **Note that the IRAC response curves are provided in units of electrons per photon! The subscript γ is supposed to remind you of this.**
- iii. Load the spectral response into IDL variables, e.g. filter_w (microns) and filter_t, like the 1st IDL line in Step 1 above. Load the template spectrum into, e.g. wave (microns) and spec_fnu (Jy).
- iv. Calculate the color correction.

```
IDL>K = spitzer_synthphot(wave, spec_fnu, filter_w,  
filter_t, 'IRAC', /colorcorr)
```

Note that the downloadable IDL code provides good agreement with the values tabulated in Section 4.4 of the IRAC Instrument Handbook for various types of science spectra. The results are consistent to within half a percent. The fact that the values are not exactly the same is likely due to differences in the interpolation and integration methods.

9.3.4 Calculate the calibrated Spitzer flux density using the following equation.

$$f_{\nu}^{\text{Spitzer}} = f_{\nu}(\lambda_{\text{eff}}) \times K$$

For the example, you could plot the Spitzer flux density:

```
IDL>fnu_Spitzer = fnu_lambda_eff * K  
IDL>plots, lambda_eff, fnu_Spitzer, psym = 8
```

Alternatively, the IDL code will also directly output the calibrated Spitzer flux density if the colorcorr keyword is not set. For example

9.4 Procedure for computing synthetic IRS Peakup photometry

9.4.1 Determine your input spectrum.

If you are going to calculate the correction using the IDL code, this should be in Janskys (i.e. energy flux per unit frequency) as a function of wavelength $f_{\nu}(\lambda)$. The input spectrum should cover the width of the IRS Peakup filter profile, so long as it has flux there. It should not be zero at the effective wavelength, and thus should generally not be a pure emission-line spectrum.

For example, you could use IDL to generate a power-law spectrum sampled at the same wavelengths as the IRS Blue Peakup filter response function (you will need the IDL Astro Library):

```

IDL>readcol, 'bluePUtrans.txt', filter_w, filter_t, format = '(d, d)',
/silent
IDL>alpha = -2. ; define the exponent of the power-law
IDL>c = 2.9979d8 * 1d6 ; speed of light in micron/s
IDL>wave = filter_w ;sample the power-law at the same wavelengths as
the filter
IDL>nu = c / wave
IDL>spec_fnu = 1e28 * nu^alpha
IDL>plot, wave, spec_fnu, xtit = 'Wavelength (microns)', $
      ytit = 'Flux Density (Jy)', charsize = 1.5, /ylog, $
      yrange = [10, 200], /ystyle ;plot the power-law spectrum
IDL>oplot, filter_w, filter_t * 50. / max(filter_t), linestyle = 1
;overplot the normalized response curve

```

9.4.2 Estimate the flux density of the input spectrum at the effective wavelength.

Given that the IRS peak-up imagers are referenced to a spectrum with $\nu S_\nu = \text{constant}$, the effective wavelength is defined as:



This is identical to equation 4.2 in the IRS Instrument Handbook. For reference, the effective wavelengths of the IRS filters are:

15.8 microns for the Blue peakup filter
 22.3 microns for the Red peakup filter

For the example with the Blue Peakup filter, you could draw the effective wavelength on the plot:

```

IDL>lambda_eff = 15.8
IDL>oplot, [lambda_eff, lambda_eff], 10.^(!y.crange), linestyle = 2

```

We can also estimate $f_\nu(\lambda_{\text{eff}})$, the flux density of the input spectrum at λ_{eff} .

```

IDL>linterp, filter_w, spec_fnu, lambda_eff, fnu_lambda_eff
IDL>plotsym, 0, 1, /fill
IDL>plots, lambda_eff, fnu_lambda_eff, psym = 8

```

9.4.3 Determine the color correction, which for the IRS is defined by the following equation.



Here, $S_v(\lambda_{\text{eff}})$ is the reference spectrum evaluated at the effective wavelength; $f_v(\lambda_{\text{eff}})$ is the template spectrum evaluated at the effective wavelength; and $R_\gamma(v)$ is the filter response function as a function of frequency in units of electrons per photon.

You can evaluate the expression above on your own by first ensuring that the response curve and the template spectrum (and reference spectrum) are on the same grid. You can choose an arbitrary normalization for the flat IRS reference spectrum (constant). However, two more convenient options are available:

- A. If your input spectrum is well-described by a blackbody or power-law over the IRS filter of interest, you can estimate the color correction from the tables in section 4.2.3.2 of the IRS Instrument Handbook.
- B. You can calculate the color correction using the IDL code as follows.
 - i. Obtain the IRS Peakup response curves, $R_\gamma(v)$. Current versions have been provided in the download directory with names **PUtrans.txt*. For updates check:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irs/calibrationfiles/>

Note that the IRS response curves are provided in units of electrons per photon! The subscript γ is supposed to remind you of this.

- ii. Load the spectral response into IDL variables, e.g. filter_w (microns) and filter_t, like the 1st IDL line in Step 1 above. Load the template spectrum into, e.g. wave (microns) and spec_fnu (Jy).
- iii. Calculate the color correction.

```
IDL>K = spitzer_synthphot(wave, spec_fnu, filter_w, filter_t, 'IRS', /colorcorr)
```

Note that the downloadable IDL code provides good agreement with the values tabulated in the tables in section 4.2.3.2 of the IRS Instrument Handbook for various types of science spectra. The results are consistent to within half a percent. The fact that the values are not exactly the same is likely due to differences in the interpolation and integration methods. Note that this IDL code computes color corrections for the IRS peak-up filters as calibrated in pipeline versions S17 and greater. It is not to be used for data reduced with older versions of the pipeline.

9.4.4 Calculate the calibrated Spitzer flux density using the following equation.

$$f_v^{\text{Spitzer}} = f_v(\lambda_{\text{eff}}) \times K$$

For the example, you could plot the Spitzer flux density:

```
IDL> fnu_Spitzer = fnu_lambda_eff * K
IDL>plots, lambda_eff, fnu_Spitzer, psym = 8
```

Alternatively, the IDL code will also directly output the calibrated Spitzer flux density if the `colorcorr` keyword is not set. For example:

```
IDL>fnu_Spitzer = spitzer_synthphot(wave, spec_fnu, filter_w,
filter_t, 'IRS')
```

9.5 Procedure for computing synthetic MIPS photometry

9.5.1 *Determine your input spectrum.*

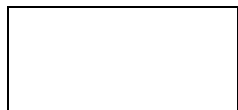
If you are going to calculate the correction using the IDL code, this should be in Janskys (i.e. energy flux per unit frequency) as a function of wavelength. The input spectrum should cover the width of the MIPS filter profile, so long as it has flux there. It should not be zero at the effective wavelength, and thus should generally not be a pure emission-line spectrum.

For example, you could use IDL to generate a 150 K blackbody spectrum sampled at the same wavelengths as the 24 microns filter response function (you will need the IDL Astro Library):

```
IDL>readcol, 'filtresponse24.txt', filter_w, filter_t, /silent
IDL>c = 2.9979d8 * 1d6 ; speed of light in micron/s
IDL>temp = 150. ; blackbody temperature in Kelvin
IDL>wave = filter_w ;sample the blackbody at the wave wavelengths as
the filter
IDL>nu = c / wave
IDL>spec_fnu = 5e9 * blackcgs(temp, nu)
IDL>plot, wave, spec_fnu, /ylog, xtit = 'Wavelength (microns)', $
          ytit = 'Flux Density (Jy)', charsize = 1.5 ;Plot the
blackbody
IDL>norm = 5. / max(filter_t)
IDL>oplot, filter_w, filter_t * norm, linestyle = 1
;overplot the normalized filter response curve
```

9.5.2 *Estimate the flux density of the input spectrum at the effective wavelength.*

For MIPS, the effective wavelength is defined as:



This is equation 4.1 in the MIPS Instrument Handbook. For reference, the effective wavelengths of the MIPS filters are:

23.68 microns
71.42 microns
155.9 microns

For the example with the MIPS 24 micron filter, you could draw the effective wavelength on the plot:

```
IDL>lambda_eff = 23.68  
IDL>oplot, [lambda_eff, lambda_eff], 10.^(!y.crange), linestyle = 2
```

Also we can estimate $f_v(\lambda_{\text{eff}})$, the flux density of the input spectrum at λ_{eff} .

```
IDL>linterp, filter_w, spec_fnu, lambda_eff, fnu_lambda_eff  
IDL>plotsym, 0, 1, /fill  
IDL>plots, lambda_eff, fnu_lambda_eff, psym = 8
```

9.5.3 Determine the color correction, which for MIPS is defined by the following equation.



Here, $S_v(\lambda_{\text{eff}})$ is the reference spectrum evaluated at the effective wavelength; $f_v(\lambda_{\text{eff}})$ is the template spectrum evaluated at the effective wavelength; and $R(\nu)$ is the filter response function as a function of frequency in units of electrons per unit energy.

Since the MIPS response is in different units than the IRAC and IRS response functions, the equation for the color correction is different.

You can evaluate the expression above on your own by first ensuring that the response curve and the template spectrum (and reference spectrum) are on the same grid. You can choose an arbitrary normalization for the 10,000 K MIPS reference spectrum. However, two more convenient options are available:

1. If your input spectrum is tabulated in Section 4.3.5 of the MIPS Instrument Handbook, you can use the color correction from the tables.
2. You can calculate the color correction using the IDL code as follows.
 - C. Obtain the appropriate MIPS response curve, $R(\nu)$. Current versions have been provided in the download directory with names *filt*.txt*. For updates check:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/mips/calibrationfiles/spectralresponse/>

Note that the MIPS response curves are provided in units of electrons per unit energy!

- D. Load the spectral response into IDL variables, e.g. filter_w (microns) and filter_t, like the 1st IDL line in Step 1 above. Load the template spectrum into, e.g. wave (microns) and spec_fnu (Jy).
- E. Calculate the color correction.

```
IDL>K = spitzer_synthphot(wave, spec_fnu, filter_w, filter_t,  
'MIPS', /colorcorr)
```

Note that the downloadable IDL code provides good agreement with the values tabulated in Section 4.3.5 of the MIPS Instrument Handbook for various types of science spectra. The results are consistent to within half a percent for 24, 70, and 160 microns. The fact that the values are not exactly the same is likely due to differences in the interpolation and integration methods.

9.5.4 Calculate the calibrated Spitzer flux density using the following equation.

$$f_v^{\text{Spitzer}} = f_v(\lambda_{\text{eff}}) \times K$$

For the example, you could plot the Spitzer flux density:

```
IDL>fnu_Spitzer = fnu_lambda_eff * K  
IDL>plots, lambda_eff, fnu_Spitzer, psym = 8
```

Alternatively, the IDL code will also directly output the calibrated Spitzer flux density if the colorcorr keyword is not set. For example:

```
IDL> fnu_Spitzer = spitzer_synthphot(wave, spec_fnu, filter_w,  
filter_t, 'MIPS')
```

Recipe 10. CUBISM: Spectrum Extraction of M51

In this recipe, we will run through the steps necessary to reduce Spitzer IRS Long-Low (LL) Mapping Data of NGC 5194 (M51).

10.1 Requirements

To follow along with this recipe, you must install the software package CUBISM (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/cubism/>). In order to run CUBISM, you must have IDL installed and make sure you have the latest version of the IDL Astronomy User's Library (<http://idlastro.gsfc.nasa.gov/>).

You will need to download the example data set using the Spitzer Archive. For this example the AORs needed are 16801536 and 16801792, all wavelengths, BCD data. See Recipe 1 for details on downloading the data. We will assume you will download and work on the data in the directory *yourdatadirectory/*.

Finally, you may wish to look at the following documents:

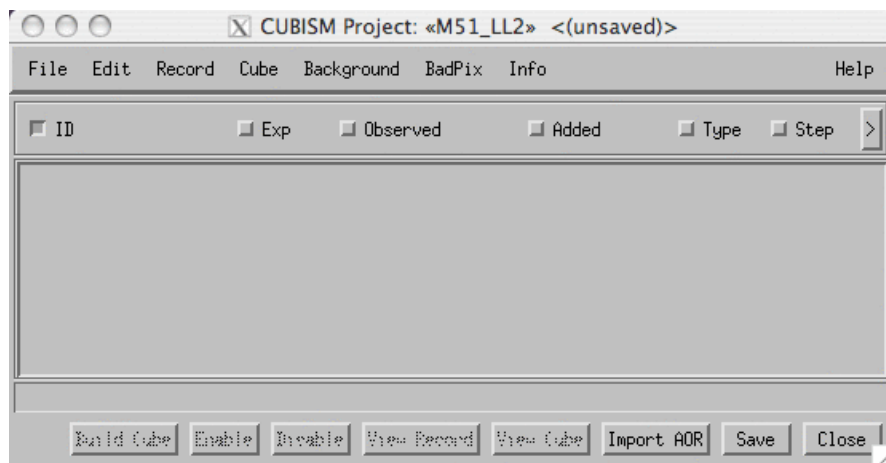
- Brunner et al. (2008, ApJ, 675..316), which discusses Spitzer IRS Maps of M51.
- The CUBISM Manual.
- The CUBISM paper by Smith et al. (2007, PASP, 119, 1133).

10.2 Create a new CUBISM project

Once you start up CUBISM, a "Load Cube Project" window will pop up. Click on "Create New Cube Project" near the bottom.



Once you give the project a name of your choice (here we choose "m51_LL2"), a CUBISM Project Window will appear:

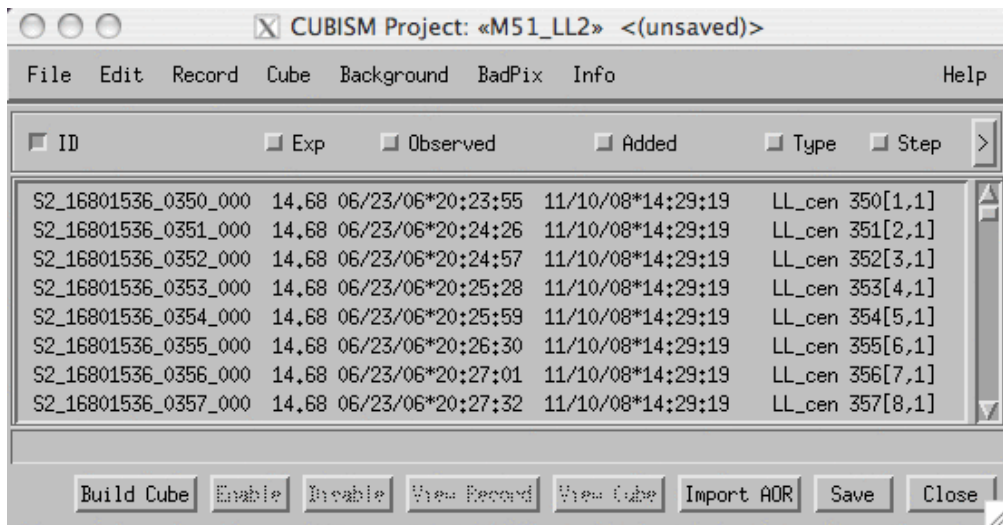


Click on "Import AOR" near the bottom of this window. Navigate to *yourdatadirectory/* and click on "OK".

CUBISM will look for all AORs in any subdirectories of *yourdatadirectory/*. Then it will ask you to verify the BCD data it finds. Check the boxes next to the "LL" options and click on "OK".

When asked, choose the most recent calibration file.

Now your Project Window should contain information about the BCD data that you just uploaded.



10.3 Visualize your AORs

CUBISM allows you to visualize the AORs you just uploaded. To try it out, first go to the main CUBISM Project Window and select "Cube"-->"Set Cube Build Order"-->2. This sets the Cube Build Order to LL2. Next select "Record"-->"Visualize AORs" in the Project Window. You will be prompted to select a Visualization Image. For this demonstration, select the 24 micron image of M51 which you downloaded previously. The file you want is called

ngc5194_mips24_image_v5-0.fits

The image you just loaded will appear in a new CubeView window. However, it will be difficult to see, so you will need to adjust the scaling. Near the top of the CubeView window, click on the histogram scaling button:



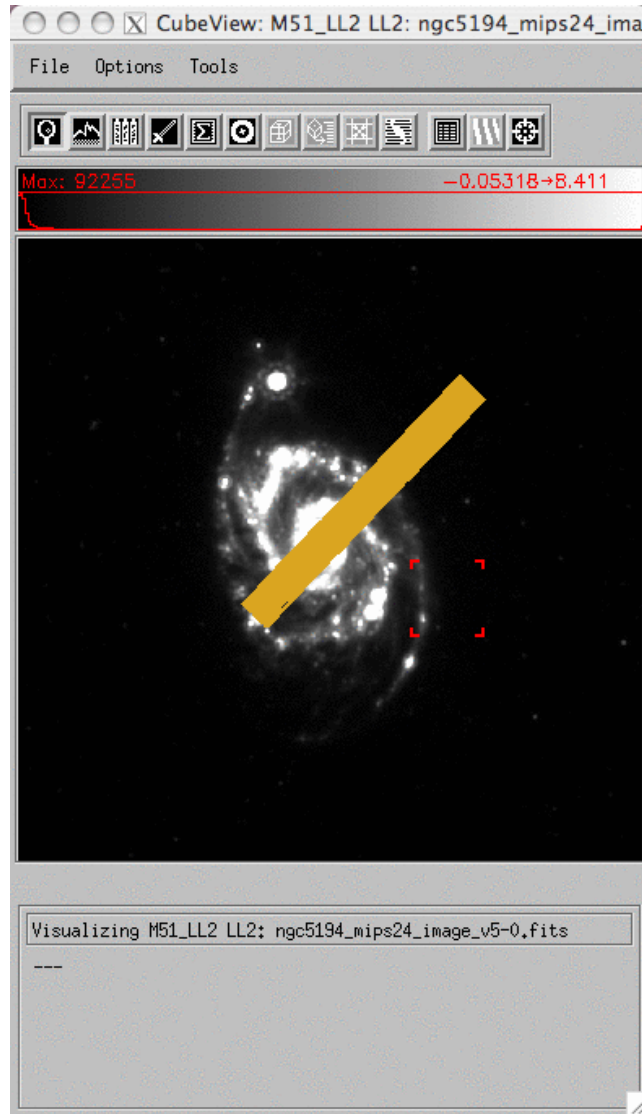
Now use your mouse to make a box anywhere in the image area. You will see the color scale change. Adjust the size and location of the box until you can see M51 properly.

Now you will notice that M51 takes up only a small part of the image area. To zoom in, click on the zoom button:



Then click anywhere on the image to zoom in and center on that spot. (Right-click to zoom out again, or command-click if you're using a Mac.)

You should see something like this:



The orange rectangles are the LL2 slits.

10.4 Background Subtraction

As can be seen in the above visualization, most of the time, the LL2 slit was place somewhere on M51. However, the LL2 observations also extend off of the galaxy. This is fortunate, because we can use these off-source positions to subtract a background from the on-source positions.

To select the background positions, simply click on the "Select BCD" Button in the CubeView Window:



and then use your cursor to click and drag over the orange rectangles that do not fall on the galaxy. They will turn white, and the corresponding entries in the Project Window will become highlighted. Since the rectangles are overlapping, it can be difficult to select the bottom layer. You can select additional records in the ProjectWindow itself. As you click on the record entries, you'll see the corresponding AORs turn white in the CubeView Window. For this example, you should select 20 records in total for the background.

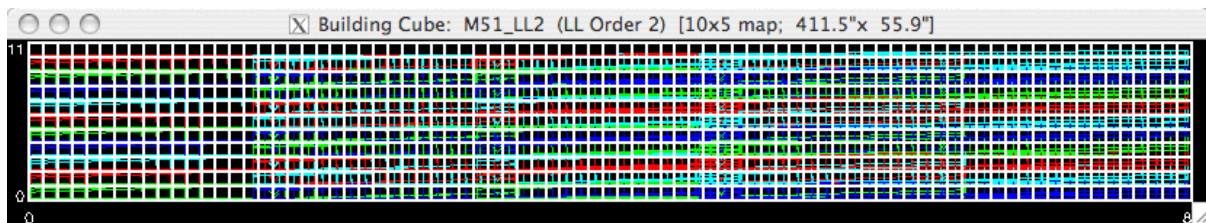
To compute a background from these selected records, go to the Project Window and click on "Background"-->"Set Background from Rec(s)". In the window that pops up, check the box next to "Average + Min/Max Trim", and click "OK".

Let's save the background records so that we don't have to go through the selection process again. In the Project Window, select "Background"-->"Save Background Rec(s)", and select the output directory.

Next disable these records so that CUBISM won't take the time to build them into the cube. In the Project Window, select "Record"-->"Disable". Now the records you selected will have a line through them to indicate that they have been disabled.

10.5 Build initial cube

In the main CUBISM Project window, click on "Build Cube" near the bottom. You will see a progress window pop up. It will look like this once the cube has finished building:

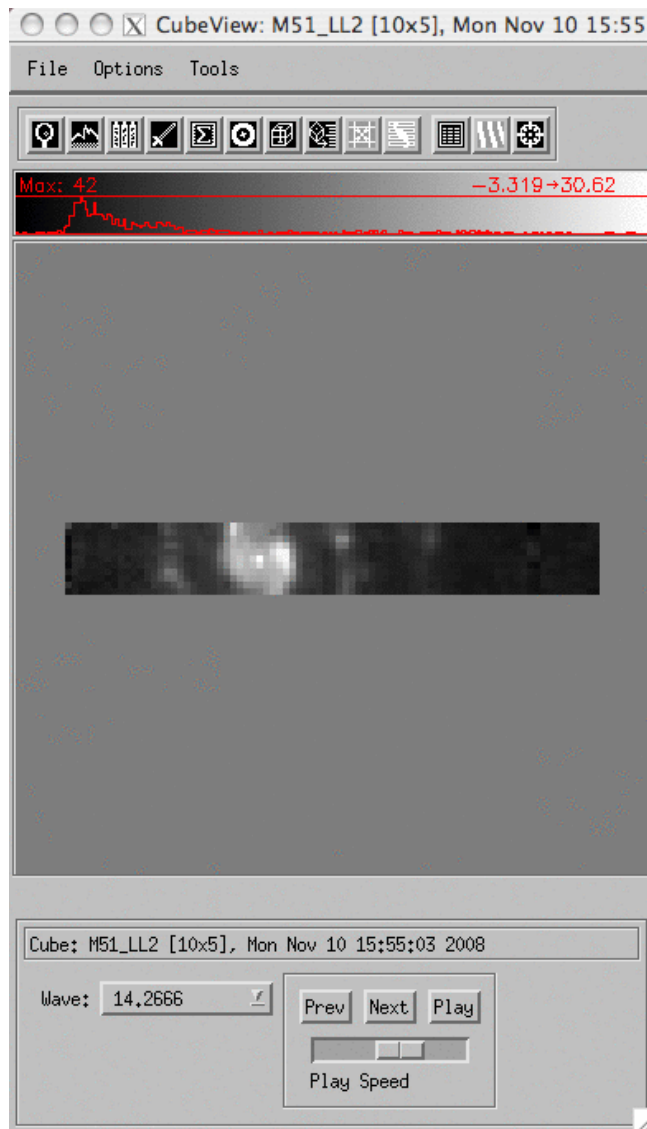


The white lines represent the output cube pixels. The colored boxes represent the input BCD pixels (or fractions thereof).

Now is a good time to save your cube. In the Project Window, select "File", "Save As" and select an output directory and filename.

10.6 Visualize cube

Now that your cube has been created, CUBISM allows you to visualize it. To do this, click on the "View Cube" button near the bottom of the CUBISM Project Window. You'll see something like this:



The image you are looking at is a picture of M51 at the wavelength 14.2666 microns. You can click the "Next" button near the bottom of the window to step through bins of increasing wavelength.

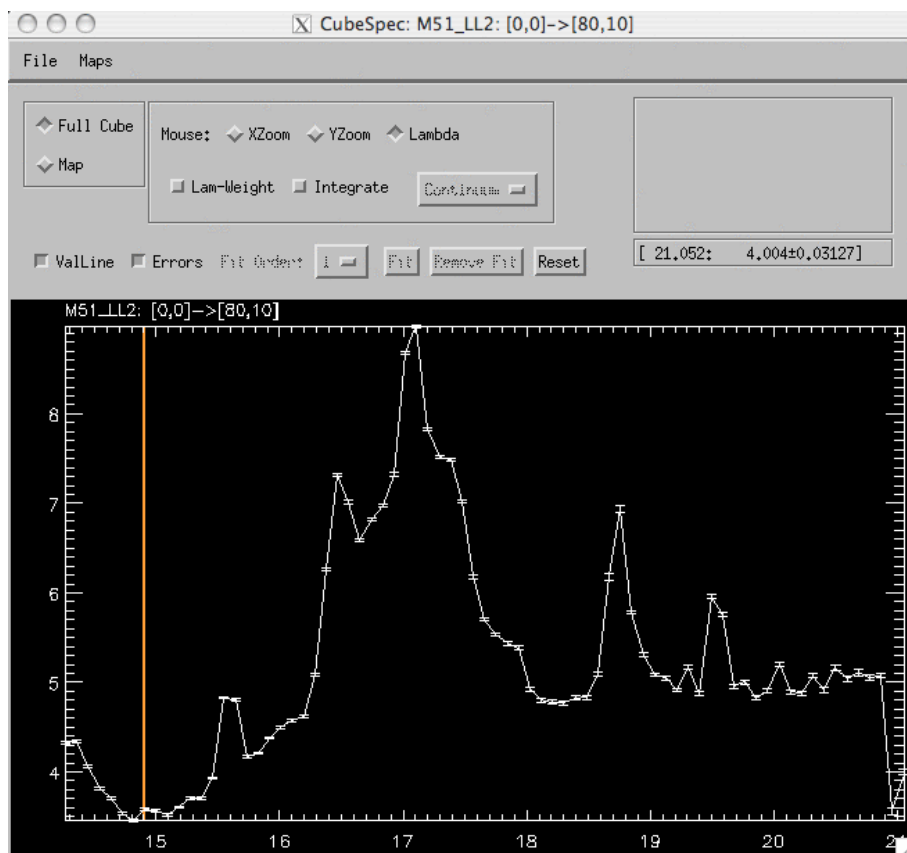
10.7 Extract spectrum

Cubism allows you to extract spectra over circular and polygonal apertures. Non-rectangular regions must be specified as ds9 regions files and then uploaded into Cubism. For more information, please see Section 6.1.2 of the Cubism manual.

To extract a spectrum over a rectangular region, in the CubeView Window, click on the "Extract Button", which looks like this:



Using your mouse, define a box in the CubeView window. This is the aperture over which your spectrum will be extracted. A "CubeSpec" Window will pop up to show you the extracted spectrum. If you selected the entire region, your spectrum should look like this:

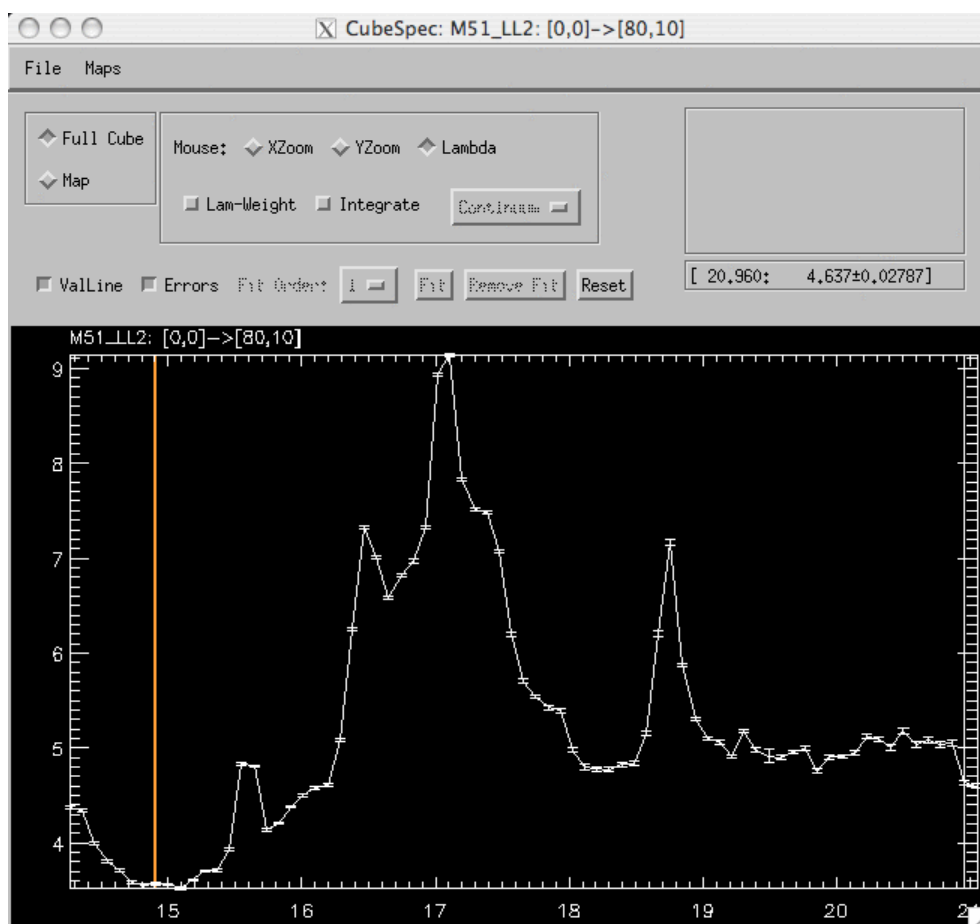


10.8 Remove bad pixels automatically

Some of the spikes in the above spectrum are real, physical features. Others are just bad pixels in the individual BCDs that need to be removed. CUBISM has a method for automatically detecting bad pixels.

In the CUBISM Project Window, choose "BadPix"-->"Auto-Gen Global Bad Pixels", then click "OK". You will be presented with two parameters which you are free to adjust: Sigma-Trim and MinBad-Frac. These parameters are described in Section 4.5 of Smith et al. (2007, PASP, 119, 1133). Briefly, for each output cube pixel, the contributing pixel residuals are tested against an estimate of the contributing pixel deviation. If a pixel is an outlier (defined by Sigma-Trim), it is tentatively marked bad. If the same pixel is so marked at least a fraction of the times (defined by MinBad-Frac) it appears in the cube, it is flagged. To avoid removing spectral lines, start with high values of Sigma-Trim and high values of MinBad-Frac. As you decrease these values, the bad pixel removal will become more aggressive, and you may start removing real features.

After you have clicked "OK" in the AutoBadPix Window, "QuickBuild" your cube. This will go much faster than the original build. Your extracted spectrum will be automatically updated:



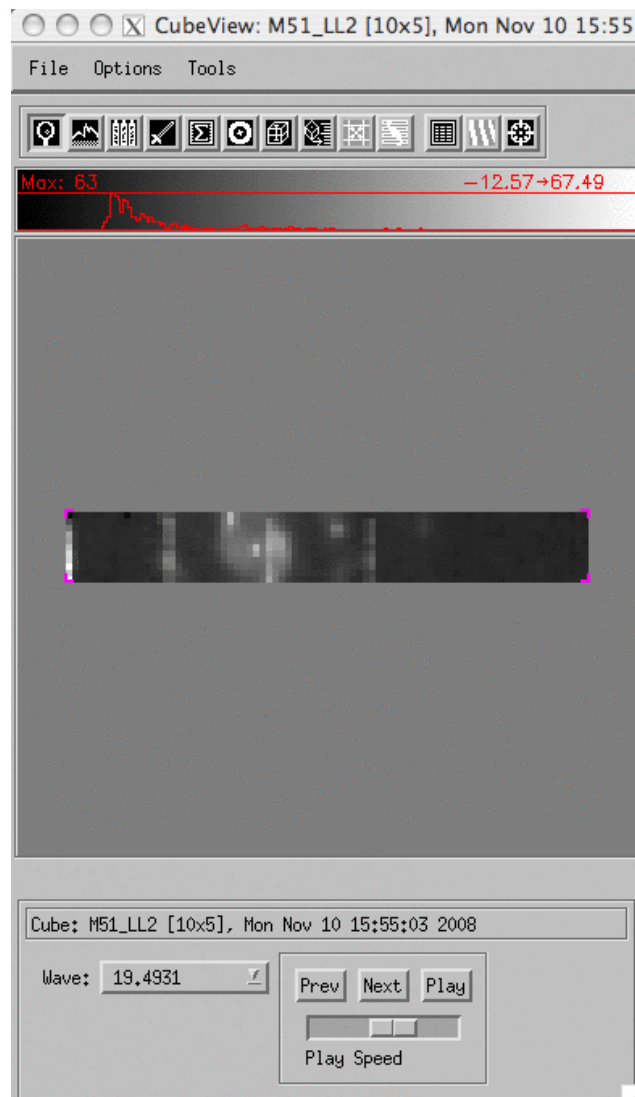
Notice that the spike at 19.5 microns went away! Also try "BadPix"-->"Auto-Gen Record Bad Pixels". Your spectrum will not change very much.

10.9 Remove bad pixels by hand

CUBISM also allows us to remove bad pixels interactively. You can do this either instead of or in addition to the automatic bad pixel rejection described in the previous section.

In our example, let's first undo the automatic bad pixel rejection so we can examine the spike at 19.5 microns in detail. In the Project Window, select "BadPix"-->"Clear All Bad Pixels", then click on "Yes". You can do a "QuickBuild" to see the spike come back.

Now let's inspect the image of M51 near the wavelength 19.5 microns. To do this, click repeatedly on the "Next" button in the CubeView Window until the wavelength reads 19.4931 microns. You should be able to see several vertical stripes in this image.



Vertical stripes like this are an indication that there is a bad pixel in one of the input BCDs. As the sky is mapped, a given bad pixel is dragged over the sky, resulting in stripes in the output cube. CUBISM's pixel backtracking tool is useful for finding the offending BCD pixels. You can activate this tool from the CubeView Window by pressing on the button that looks like this:



Once you've clicked on this button, you can place your cursor over any cube pixel in the CubeView Window to see which input BCD pixels contributed to it. Try clicking on a cube pixel in the rightmost vertical strip. You will see a green 'x' appear on that pixel and the BackTracking Window will freeze, looking something like this:

BCD	Pix	Frac	Val	Back	(Val-Back)	Flag
S2_16801536_0360_0000	(93, 18)	0,199	55,9±10,3	56,0±2,45	-0,0781±10,6	
	(93, 19)	0,312	255,±15,0	247,±3,79	7,39±15,5	
	(94, 18)	0,112	169,±22,0	124,±3,04	44,1±22,2	
	(94, 19)	0,171	64,8±10,6	55,6±2,43	9,18±10,8	
S2_16801536_0379_0000	(77, 19)	0,157	67,0±10,6	57,6±2,60	9,47±10,9	
	(78, 19)	0,644	57,1±10,2	55,9±2,51	1,24±10,5	
S2_16801536_0380_0000	(62, 19)	0,481	400,±45,0	-1,24E+03±14,4	1,64E+03±47,2	[7]
	(62, 20)	0,0962	2,64E+03±75,7	2,38E+03±17,1	262,±77,6	[7]

You can see that pixels (62, 19) and (62,20) in one BCD is very large, and both have bmask flags of 7. Back in the CubeView Window, right-click on the green 'x' you made, and it will disappear. Once again, you can run the cursor over any pixel and the BackTracking Window will update. Running the cursor over cube pixels within all three stripes, you will see that these particular BCD pixels are bad in several of them.

To mark these pixels as bad, do the following:

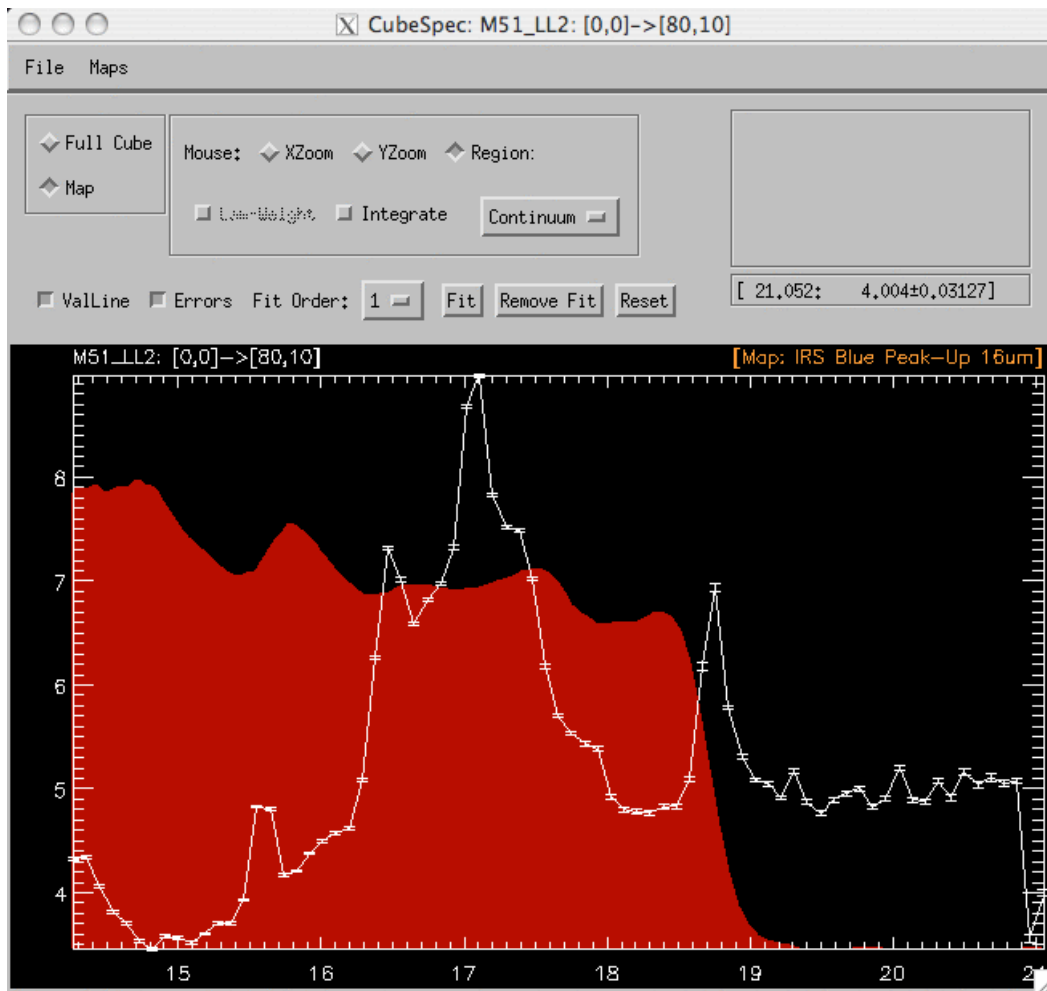
1. Click on one of the cube pixels in the stripe in the CubeView window.
2. Go to the Backtracking window, and click on the line for the pixel (62,19) to highlight it.
3. Right-click (command-click for Mac users) on the highlighted line.
4. You'll be asked whether to set the pixel to a "global", "this record" or "these records" bad pixel. In this case, the pixel is bad throughout, so we will set it to a global bad pixel. Check the Bad Pixel (Global) box. The flag will change to "BP[7]".
5. Repeat for pixel (62,20).

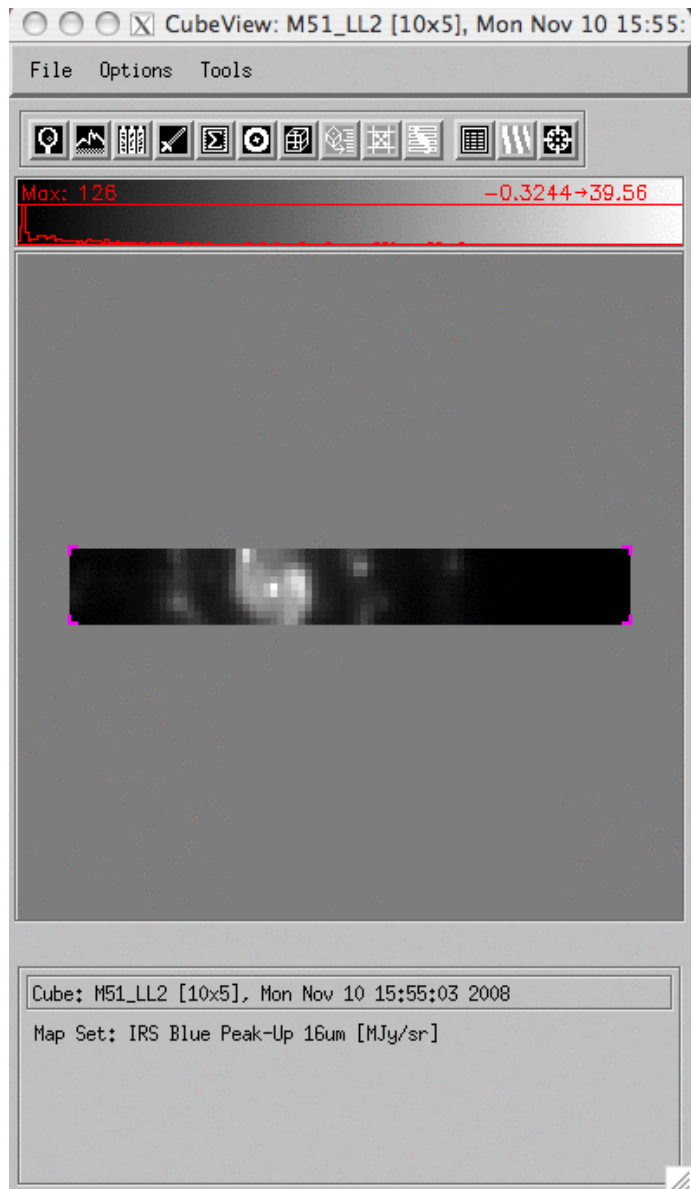
If you do a quick build of the cube, you will see the negative spike disappear in the extracted spectrum, and the stripes disappear in the CubeView Window at that wavelength.

If you end up marking a lot of bad pixels by hand, you will probably want to save your bad pixel list. To do this, go to the CUBISM Project Window and click on "BadPix"-->"Save Bad Pixels".

10.10 Make a filter-specific map

In addition to making a map at a given wavelength, CUBISM can make a map using a given filter. In the CubeSpec Window, select "Maps". You will see that several options are available. None of these fall entirely within the LL2 wavelength coverage, but you can select "IRS Blue Peak-Up 16um" to see what happens. The CubeSpec window will show the 16 microns filter curve in red, and the CubeView Window will now show a map integrated over that curve:





10.11 Save your work

To save the cube project again (cpj file), go to the CUBISM Project Window, click on "File"-->"Save".

You may also want to save the cube as a fits file. To do this, go the CUBISM Project Window and click on "File"-->"Write Fits Cube".

Recipe 11. Dark_Settle: Correct Dark Currents for HR 6606

Data taken with the Long-High (LH) module of Spitzer's IRS sometimes exhibit time-dependent dark currents. Because this dark current is not uniform on the array, it can lead to order tilting or scalloping, as well as order mismatch. (Please see the IRS Instrument Handbook for more information on LH Order Tilts). In this recipe, we demonstrate how to recognize this effect in your data, as well as how to correct your Basic Calibrated Data files (BCDs) using the contributed IDL software `dark_settle`. In addition, we introduce the user to the batch mode option in SPICE. This mode allows the user to run multiple input BCDs (and associated mask and uncertainty files) through a single SPICE flow.

11.1 Requirements

You will need a licensed version of IDL; `dark_settle` works only with IDL versions 6.2 and greater. Also, download the IDL software package `dark_settle` (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/darksettle/>) and the IRS spectral extraction software SPICE (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/spice/>), or you may also use SMART (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/contributed/irs/smart/>).

To follow along with this example, you will need to use the Spitzer archive to download the example data set. See Recipe 1 for details on how to do that. You will need the BCD and Calibration data for the AOR 20072448. The target is HR 6606 and it is part of an IRS calibration program. There should be eight BCD FITS files associated with this Long-High IRS observation.

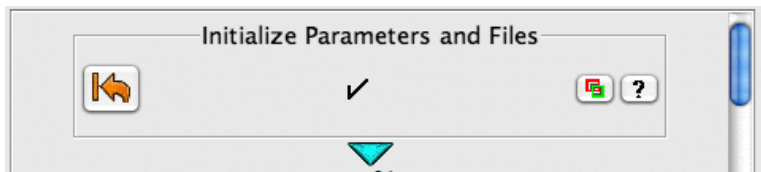
11.2 First-pass extraction

The first step is to perform a simple extraction on your eight BCD files. Ideally, one would first perform background subtraction on the BCD files, as well as use IRSCLEAN (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/irsclean/>) to remove rogue pixels in the background-subtracted BCD data. However, for this example, we wish to focus on the effect of running `dark_settle` alone. Therefore, we use SPICE to extract 1D spectra from the BCD files directly. Since this is a bright point source, we can use the regular point source extraction flow in SPICE. To start, we will use all the default parameters.

We must perform this extraction on several BCDs. Therefore, we can choose to run SPICE multiple times, or we can use the batch mode option in SPICE to run several input BCDs through the same extraction process all at once. Below, we outline the steps for both approaches.

11.2.1 Running SPICE on several BCDs, one at a time:

1. Start SPICE.
2. "File"-->"Open SPICE Generic Template"-->"Point Source with Regular Extract"-->"OK "
3. For the input Image File, select the first of the BCDs, *SPITZER_S3_20072448_0012_0000_4_bcd.fits*. SPICE will automatically select the correct mask and uncertainty files.
4. Choose your output directory, e.g. *first_pass_output*.
5. Run Spice. Simply click on the green button near the top.
6. Rewind the SPICE flow to the beginning by scrolling up to the beginning of the flow and clicking on the left-pointing arrow. It should look like this:



7. Repeat steps 3 (substituting names), 5, and 6 for the remaining spectra. Your extracted spectra will be written out as:
*/dark.settle/r20072448/ch3/bcd/first_pass_output/SPITZER_S3_20072448_0012_000*_4_bcd.spect.tbl.*

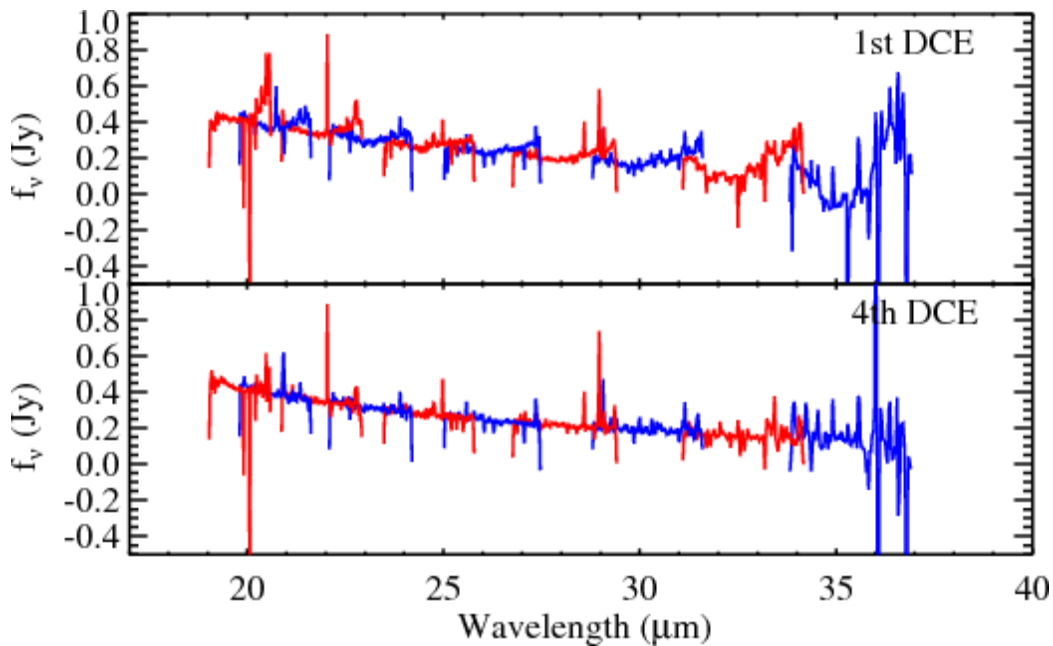
11.2.2 Running SPICE on several BCDs, using batch mode:

1. Start SPICE.
2. "File"-->"Open Batch SPICE Generic Template"-->"Point Source - Regular Extract - Hi Res Option"-->"OK".
3. Set "Current Output Directory" to *first_pass_output*.
4. Make your batch input list. This can be done in two ways:
 1. Choosing and Clicking. Set "Image File" to *SPITZER_S3_20072448_0012_0000_4_bcd.fits*. SPICE will automatically find the corresponding entries for "Mask File" and "Uncertainty File". Click on "Add to List", and watch these three files appear in the "Batch List" inset. Repeat this step for the remaining BCD files.
 2. Uploading a file. Create a file that contains all the input BCD, mask file, and uncertainty files. Each triplet should occupy a single line of the file. In each line, the BCD file should appear first, followed by the mask file and the uncertainty file, respectively. These files should be separated by commas.

5. Once all of your input files appear in the "Batch List" inset, click on the green arrow at the top of the SPICE window to run your batch flow. The extracted spectra will have the same filenames as they would if you ran the BCDs through SPICE one at a time, but they will appear in numerical subdirectories ('1', '2', '3', etc.) of your chosen output directory.

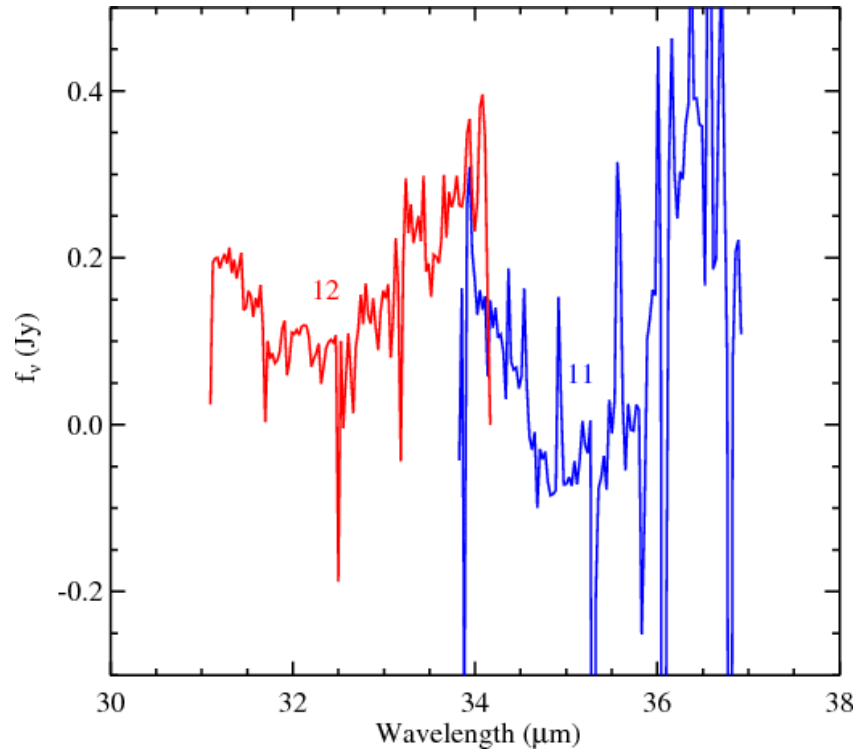
11.3 Time-Dependent Darks: Recognizing the Problem

In the last section we demonstrated how to extract 1D spectra from your 2D BCDs. Below we show the spectra corresponding to the 1st and the 4th DCEs:



You can see that the first DCE has scalloped orders, which look like the letter "U". The scalloping subsides with each DCE, and one can see that they are significantly diminished by the 4th DCE. Note that the large spikes are due to bad pixels that could be removed with the software package IRSCLEAN. However, for the purposes of this demonstration, we are more interested in correcting the continua of the spectra in each order.

The image below zooms in on orders 11 and 12 of the 1st DCE above. The "U"-shaped orders are very clear.



Scalloping like this can be due to a time-dependent dark current (see the IRS Instrument Handbook). Excess flux is not distributed evenly across the array, causing the scalloping. Below we show the 2D spectra of the first DCE and the fourth DCE. It is easy to see how non-uniform the background is in the first DCE, compared to that in the fourth DCE.



Left is the first DCE and the right image is the fourth DCE.

11.4 Mitigation: Running `dark_settle`

The `dark_settle` software operates on a set of IRS BCD and calibration images (*cal*) from a particular AOR. First, the software uses the *cal* files to undo the flatfield correction that has been applied to the BCD data by the SSC IRS pipeline. This is because the unilluminated interorder regions, from which the `dark_settle` correction is derived, have not been flatfielded, so the correction needs to be applied to the unflatfielded data. Next, the correction is applied: For a given row of a given BCD, `dark_settle` computes an interorder mean, smoothed along the column. It then subtracts this mean from all the data in the row. In this way, the interorder region for each row is set to zero. The algorithm is based on the assumption that, in the absence of settling, each BCD would have an interorder flux of zero. Finally, the flatfield correction is re-applied.

For more documentation on `dark_settle`, see the header at the top of the file *dark_settle.pro*. You can see this header by typing `dark_settle, /help` at the IDL prompt.

To run `dark_settle`, you need to make a list of BCD files in your AOR. This list can be in any order. For the purposes of this example, we will call this list *r20072448/ch3/bcd/bcdlist.txt*. It should look like this:

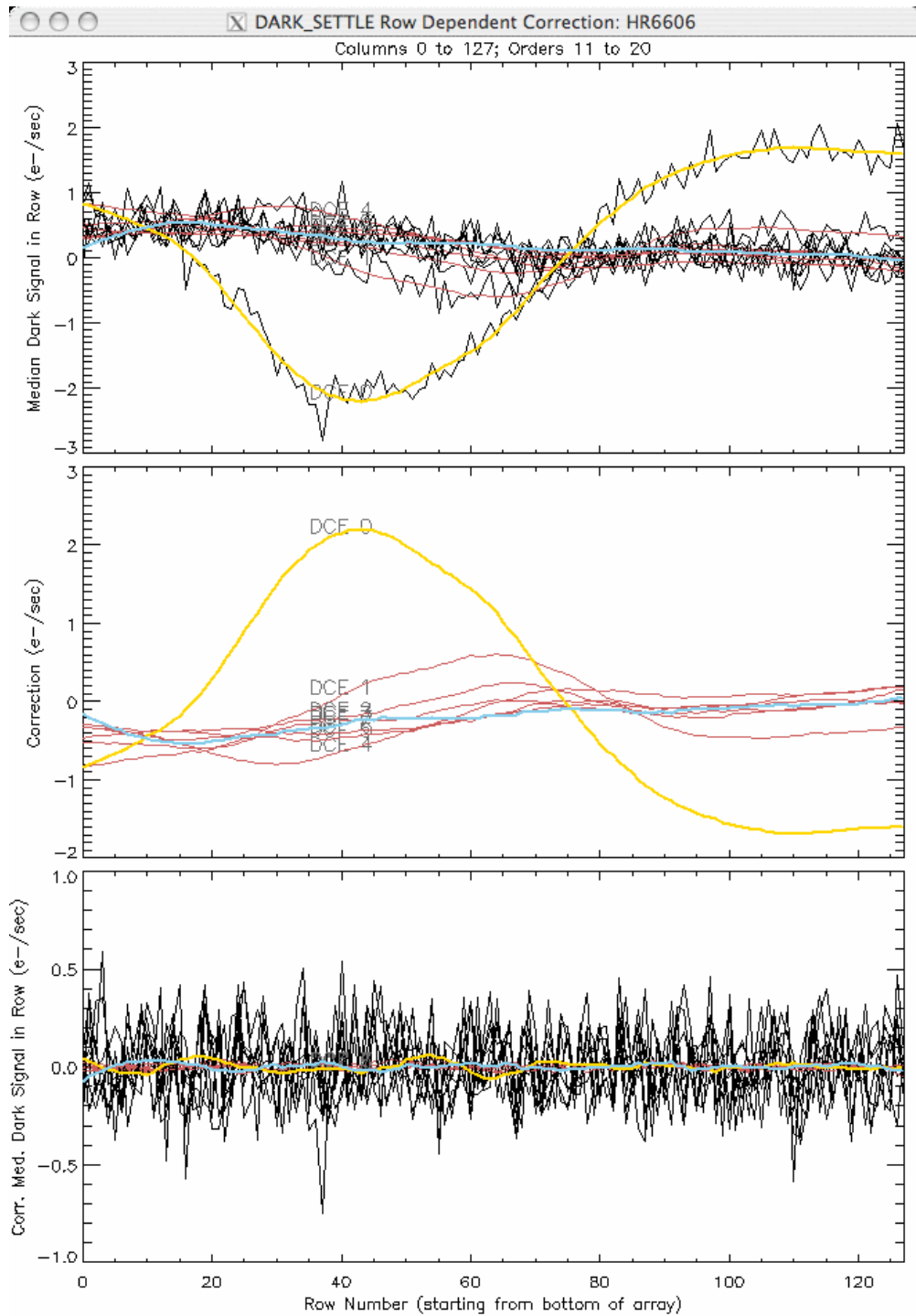
```
SPITZER_S3_20072448_0012_0000_4_bcd.fits
SPITZER_S3_20072448_0012_0001_4_bcd.fits
SPITZER_S3_20072448_0012_0002_4_bcd.fits
SPITZER_S3_20072448_0012_0003_4_bcd.fits
SPITZER_S3_20072448_0013_0000_4_bcd.fits
SPITZER_S3_20072448_0013_0001_4_bcd.fits
SPITZER_S3_20072448_0013_0002_4_bcd.fits
SPITZER_S3_20072448_0013_0003_4_bcd.fits
```

Note that unless you wish to run `dark_settle` from the `bcd` directory, you must include the full path in *bcdlist.txt*.

```
IDL> dark_settle
```

A window will pop up directing you to select a set of images or list files to process. Select the list you made before (*r20072448/ch3/bcd/bcdlist.txt*). For additional input options, such as using unix-style wildcards, please see the documentation in the `dark_settle.pro` file header, which you can also access by typing `dark_settle, /help` at the IDL prompt.

The following window will pop up:



The yellow line represents the first DCE (0) and the blue line represents the last DCE (7). In the first panel, we can see that the dark signal depends strongly on the row number, consistent with what we see in the 2D spectrum above. The second panel shows the correction required to remove this variation, while the third panel shows the corrected dark signal.

You are finished running `dark_settle`, and can dismiss the window at your leisure.

11.5 The outputs of dark_settle

The corrected file names will have the string "dks" appended to the front of the file type. If the file is *[...]_bcd.fits*

the corrected version will be

[...]_dks_bcd.fits.

In our example, the outputs are:

```
SPITZER_S3_20072448_0012_0000_4_dks_bcd.fits
SPITZER_S3_20072448_0012_0001_4_dks_bcd.fits
SPITZER_S3_20072448_0012_0002_4_dks_bcd.fits
SPITZER_S3_20072448_0012_0003_4_dks_bcd.fits
SPITZER_S3_20072448_0013_0000_4_dks_bcd.fits
SPITZER_S3_20072448_0013_0001_4_dks_bcd.fits
SPITZER_S3_20072448_0013_0002_4_dks_bcd.fits
SPITZER_S3_20072448_0013_0003_4_dks_bcd.fits
```

11.6 Is our problem solved?

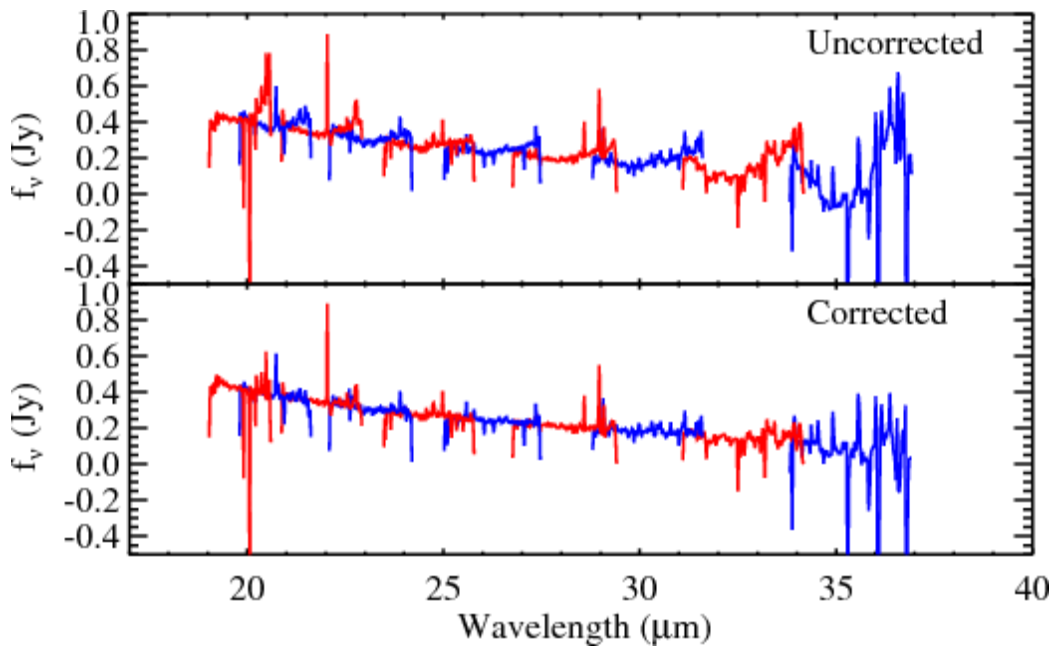
Now we must test whether dark_settle mitigated our problem. First, let's compare the uncorrected and corrected 2D spectra for the first DCE (the one with the largest correction).



Left is the uncorrected first DCE and to the right is the corrected first DCE. The corrected DCE clearly has a more uniform background.

We can also use SPICE to do a simple point-source extraction, as before. Ideally, one would first perform background subtraction on the *dks_bcd* files, as well as use IRSCLEAN to remove rogue pixels in the

background-subtracted *dks* data. However, for this example, we wish to illustrate the effect of running `dark_settle` alone. Therefore, we use SPICE to extract 1D spectra from the *dks* files directly. In the following plot, we show the uncorrected spectrum from the 1st DCE in the top panel (same as the top panel in the first figure in Section 8.3) and the corrected spectrum in the bottom panel. You can see that the scalloping is significantly reduced in the bottom panel. Based on the previous figure, and on the next figure, `dark_settle` has successfully mitigated our problem with time-dependent dark current.



A table file for the 1D spectrum can be found here:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/SPITZER_S3_20072448_0012_0000_4_dks_bcd.spect.tbl

Recipe 12. SPICE: Command-line Extraction of HH 46/47

In this recipe, we demonstrate how to use the command-line version of SPICE to extract a Spitzer IRS short-low (SL) spectrum of Herbig-Haro 46/47.

12.1 Requirements

You will need to use two pieces of software to follow this recipe:

1. Spitzer IRS Custom Extraction (SPICE):

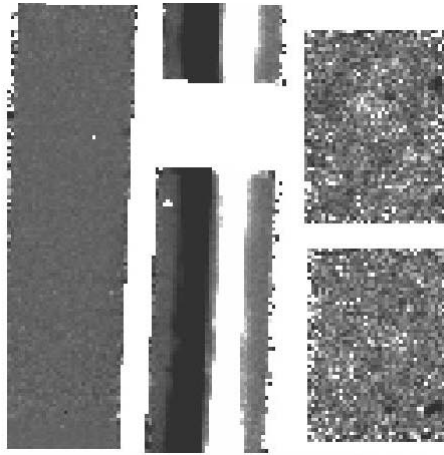
<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/spice/>

2. The Spitzer archive interface software, and data from AOR 7130112 (all wavelengths, post-BCD data). See Recipe 1 for instructions on downloading the data for the AOR. For this specific example we will use just the short-low data. Specifically, within the *r7130112/ch0/pbcd/* directory, using the tool "imhead" (part of the WCS tools package: <http://tdc-www.harvard.edu/software/wcstools/index.html>), you can check the keyword "FOVNAME" in the **bksub.fits*:

```
unix% ls *bksub.fits
SPITZER_S0_7130112_0002_6_A14567688_bksub.fits
SPITZER_S0_7130112_0002_6_A14567695_bksub.fits
SPITZER_S0_7130112_0002_6_A14567703_bksub.fits
SPITZER_S0_7130112_0002_6_A14567711_bksub.fits

unix% imhead *bksub.fits | grep 'FOVNAME'
FOVNAME = 'IRS_Short-Lo_2nd_Order_1st_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_2nd_Order_2nd_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_1st_Order_1st_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_1st_Order_2nd_Position' / Field of View Name
```

We will extract a spectrum from the first image in the list, which is shown below. The dark pixels have large values, and the white pixels have small values (or NaN between the orders).



Note that you see two spectral traces. The one on the left is positive, and the one on the right is negative. The left trace (1st nod position) has been background-subtracted using the the 2nd nod position. This leaves a negative trace in the 2nd nod position. You will be extracting the left, positive, nod 1 trace.

12.2 Why use the command-line version of SPICE?

If you have only a handful of spectra to extract, we suggest that you use the GUI version of SPICE. The GUI version provides visualization features (2D spectrum display, profile plots, ridge overlays, etc.) which can help you ensure that your extraction is going as desired. You can find examples of how to run the SPICE GUI later in this Cookbook.

If you have a large number of spectra that must all be reduced in an identical way, one option is to use the SPICE GUI in batch mode. You can find an example of how to do this embedded in the DARK_SETTLE recipe in this Cookbook (Recipe 11).

If you have a huge number of spectra and/or limited computer memory, batch mode may be insufficient to reduce all of your spectra in one batch. In that case, you may either choose to run multiple batches, or to use the command-line version of SPICE to write a script to reduce your spectra.

If you choose to write a script, we recommend that you first use the GUI on a few of your spectra to experiment with your extraction parameters. You can then use the command line to script the remainder of your extractions using the same, optimized parameters. Every time the GUI runs, it writes out the corresponding command-line parameters. Check the *spice*log* files in your *.spot/* directory. For each module, the input parameters are listed on separate lines for improved readability. To execute these on the command line, you will have to reformat them into one line. In addition, **you must make sure that the SPICE binaries are in your path.** (On a Mac, these binaries can be found in the *bin/* subdirectory of your installation. For Solaris they can be found in the *platform/sun/bin* subdirectory.)

Below we go through an example of how you would execute these commands to perform a regular, standard-width, point-source extraction on a SL1 IRS spectrum. You will need to modify these parameters if you are using a different module or if you wish to perform a custom-width extraction. We

strongly recommend that you make your modifications using the GUI first, and then copy those parameters when constructing your command-line syntax.

12.3 Run the PROFILE Module

The first step in the extraction process is to compute and plot the mean spatial flux profile across all user-selected orders. This will help us to identify the extraction window in the next step.

To run PROFILE from the command-line, first make sure that the PROFILE binary, found in the *bin/* subdirectory of your installation, is in your path. Then issue the following (long) one-line command in a directory which contains your *bksub.fits* file and its associated mask (*bkmsk.fits*) file. Please note that you will have to modify the command below to provide the full path to the *cal/* directory in your SPICE installation.

```
unix% profile -i SPITZER_S0_7130112_0002_6_A14567688_bksub.fits -b
SPITZER_S0_7130112_0002_6_A14567689_bkmsk.fits -fb 28800 -o
SPITZER_S0_7130112_0002_6_A14567688_bksub.profile.tbl -q
SPITZER_S0_7130112_0002_6_A14567688_bksub.profile.qa -t
cal/C15.0PRE25/b0_psf_fov.tbl -w cal/C15.0PRE25/b0_wavsamp.tbl -c 1000
```

You are free to modify the values for the above input flags. You can type 'profile' with no argument at the prompt for a brief description of each possible input parameter. Also, you may wish to inspect the contents of the file *spice/cdf/*/*_settings.prop* to help you decide which values to give the input flags for data with a particular CAL_SET. The flags used above are described below.

- *i* - The name of the input 2D spectrum from which you wish to extract a 1D spectrum.
- *b* - The mask file associated with the input spectrum.
- *fb* - Fatal bit pattern for the mask file. Set this to the decimal representation of the fatal bit flags. Pixels with the specified flags will be excluded from SPICE extraction. See Appendix B of the SPICE Manual for instructions on how to calculate the fatal bit pattern. Default value is 18432.
- - The name of the output table which will hold the wavelength-collapsed average spatial profile for the selected orders. Default value is *(input file).profile.tbl*.
- *q* - The name of the output qa file. Default value is *(input file).profile.qa*. This file tells you six things about the profile:
 - profMaxFlux - max value of dn for all cuts
 - profLocMaxFlux - location (%) of max dn cut
 - profMaxStdev - max value of stdev for all cuts
 - profLocMaxStdev - location (%) of max stdev cut

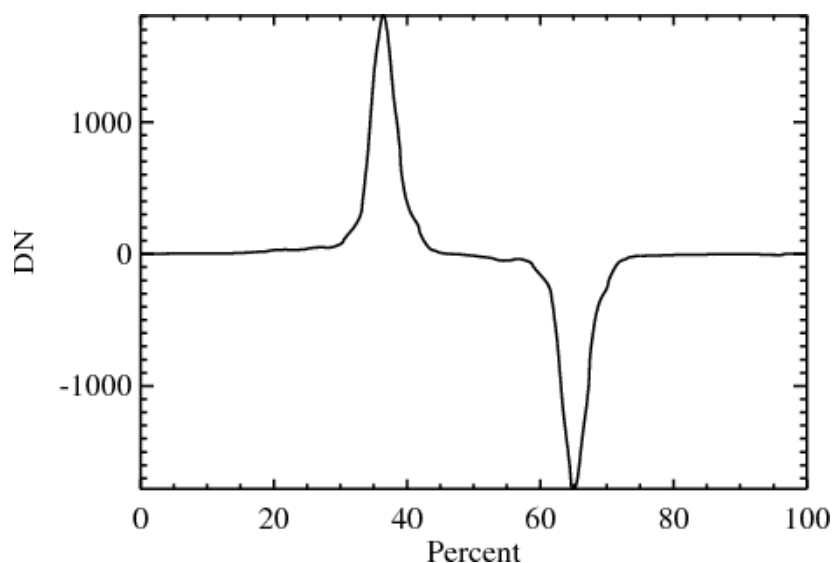
profMaxRStdev - max value of stdev/dn for all cuts
profLocMaxRStdev - location of max stdev/dn cut

- *t* - The name of the **psf_fov.tbl* calibration table which codes the order to be extracted for each FOVID, the default ridge location, and the extraction aperture width at a fiducial wavelength.
- *w* - The name of the **wavesamp.tbl* file. This file specifies the location of the spectral orders on the array in x-y coordinates. It consists of pseudorectangles which describe the fractional pixels which comprise each wavelength in the spectrum. You should choose the appropriate file in the *cal/* subdirectory of your SPICE installation. The file you choose should match both the Spitzer pipeline version with which your data were processed and the IRS module with which you are working.
- *c* - The PROFILE module divides each pseudorectangle into this number of cuts and integrates the signal within each cut. Allowed values are integers between 1 and 2000. We recommend that users keep this at the default value (1000 for SL and LL; 200 for SH and LH) adopted by the SPICE GUI.

The important output from issuing this command is

SPITZER_S0_7130112_0002_6_A14567688_bksub.profile.tbl, a table of the wavelength-collapsed average spatial profile for the selected orders.

(http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/SPITZER_S0_7130112_0002_6_A14567688_bksub.profile.tbl), Below is a plot of the data in this table.



Here again we see two features, a positive one on the left and a negative one on the right. Recall that the left peak represents the background-subtracted nod 1 spectrum, while the right-hand dip represents the sky background in the nod 1 image minus the nod 2 spectrum.

12.4 Run the RIDGE Module

The next step in the extraction process is to use the RIDGE module to identify the location for point source extraction by identifying the peak in the PROFILE output, or by using a user-specified fractional location along the slit.

To run the RIDGE module from the command line, try the following. Please note that you will have to modify the command below to provide the full path to the *cal/* directory in your SPICE installation.

```
unix% ridge -p SPITZER_S0_7130112_0002_6_A14567688_bksub.profile.tbl -
f cal/C15.0PRE25/b0_psf_fov.tbl -m 5.0 -s 3.0 -g 25.0 -o
SPITZER_S0_7130112_0002_6_A14567688_bksub.ridge.tbl -i
cal/C15.0PRE25/b0_wavsamp.tbl
```

You are free to modify the values for the above input flags. You can type 'ridge' with no argument at the prompt for a brief description of each possible input parameter. Also, you may wish to inspect the contents of the file *spice/cdf/**/_settings.prop* to help you decide which values to give the input flags for data with a particular CAL_SET. The flags used above are described below.

- *p* - The output table from running the PROFILE module in the previous step.
- *f* - The **fov.tbl* file, which codes the order to be extracted for each FOVID, the default ridge location, and the extraction aperture width at a fiducial wavelength.
- *m* - The number of points to use for median filtering (LO-RES only). Must be ODD.
- *s* - The number of sigma to require the peak to be above the mean (LO-RES only). Must be between 0.0 and 10 (inclusive). The default is 5.
- *g* - The width, given in plus-or-minus percent of the array in the cross-dispersion direction, to be analyzed, centered on expected position (LO-RES only). Allowed values are floats from 0.0 to 100, default is 100.0. e.g. if you wish to include 25% of the array either side of the profile center, giving you a total extraction width of 50% of the array, you would set a value of 25.0.
- - The output table containing the x,y position of the peak for each order and each wavelength.
- *i* - The name of the **wavsamp.tbl* file. This file specifies the location of the spectral orders on the array in x-y coordinates. It consists of pseudorectangles which describe the fractional pixels which comprise each wavelength in the spectrum. You should choose the appropriate file in the *cal/* subdirectory of your SPICE installation. The file you choose should match both the Spitzer pipeline version with which your data were processed and the IRS module with which you are working.

The important output from issuing this command is

SPITZER_S0_7130112_0002_6_A14567688_bksub.ridge.tbl, a table containing the x,y position of the peak for each order and each wavelength.

(http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/SPITZER_S0_7130112_0002_6_A14567688_bksub.ridge.tbl).

12.5 Run the EXTRACT Module

Here we demonstrate two options for performing the extraction: regular and optimal (flux-weighted).

To perform a regular extraction within a user-specified window, try the following. Please note that you will have to modify the command below to provide the full path to the *cal/* directory in your SPICE installation.

```
unix% extract -i SPITZER_S0_7130112_0002_6_A14567688_bksub.fits -o
SPITZER_S0_7130112_0002_6_A14567688_bksub.extract.tbl -b
SPITZER_S0_7130112_0002_6_A14567689_bkmsk.fits -fix 2 -nanDrop 1 -f
29056 -e SPITZER_S0_7130112_0002_6_A14567690_bkunc.fits -r
SPITZER_S0_7130112_0002_6_A14567688_bksub.ridge.tbl -ord 0 -p
cal/C15.0PRE25/b0_psf_fov.tbl -height1 0 -w 4.0 -l 6.0 -full 0 -norm 0
```

You are free to modify the values for the above input flags. You can type 'extract' with no argument at the prompt for a brief description of each possible input parameter. Also, you may wish to inspect the contents of the file *spice/cdf/*/*_settings.prop* to help you decide which values to give the input flags for data with a particular CAL_SET. The flags used above are described below.

- *i* - The name of the input 2D spectrum from which you wish to extract a 1D spectrum.
- *-* - The name of the output table, which will contain the 1D spectrum in units of electrons per second.
- *b* - The name of the mask file that goes with the input 2D spectrum.
- *fix* - This controls if the NaNs should be replaced with approximated values before doing extraction. A value of 0 means "don't replace". An integer, *N*, means that for each NaN pixel, the *N* rows above and *N* rows below are averaged column by column within the order, and then a line is fit to these column averages to approximate the NaN. Default is 2.
- *nanDrop* - Wavelength bins with NaNs will be removed from the output spectrum if this is set to 1. Default is 0, with the NaN fluxes set to -9999.

- f - Fatal bit pattern for BMask. See Appendix B in the SPICE Manual for the allowed values and a description of how to calculate the fatal bit pattern. Default = 16384.
- e - The name of the uncertainty file that goes with the input 2D spectrum.
- r - The name of the output table from running the RIDGE module in the previous step.
- ord - This refers to the spectral order, and is used for low-resolution extractions only. You can set this to:
 - 0 (as targeted; default)
 - 1
 - 2 (2+bonus)
 - 99 (1+2+bonus)
- p - The name of the **psf_fov.tbl* calibration table which codes the order to be extracted for each FOVID, the default ridge location, and the extraction aperture width at a fiducial wavelength.
- height1 - Sets the wavesamp height. Set to 1 to force wavsamp height to 1 pixel; set to 0 to use actual height. Default = 0.
- w - Width (in pixels) of the extraction aperture at the reference wavelength. Default for HI-RES is to use the full width of the order, and for LO-RES to use the value given in the *psf_fov.tbl* table specified by the "-p" flag.
- l - Reference wavelength (in microns) of the aperture width specified by the "-w" flag. A value of 0 selects a constant width extraction aperture. Default is to use the value given in the *psf_fov.tbl* file specified by the "-p" flag.
- full - Extraction type; 0=sub_slit and 1=full_slit. Default is sub-slit unless all of the following are true: (a) the "-w" flag is not explicitly set; (b) this is not a HI-RES observation; (c) this is not a LO-RES center position observation.
- norm - Normalize the fluxes. Default is no normalization.
 - 0 = no normalization
 - 1 = divide by height of wavsamp rectangle
 - 2 = divide by area of wavsamp rectangle

Set to 1 for S18.7 SL, LL, LH.
 Set to 0 for S18.7 SH.
 Set to 1 for S17.2 LH.
 Set to 0 for S17.2 SL, LL, SH.
 Set to 0 for all other CAL_SET versions.

The important output from issuing this command is

SPITZER_S0_7130112_0002_6_A14567688_bksub.extract.tbl

(http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/SPITZER_S0_7130112_0002_6_A14567688_bksub.extract.tbl), the one-dimensional extracted spectrum, in units of electrons per second.

To perform optimum extraction, try the following. Please note that you will have to modify the command below to provide the full path to the *cal/* directory in your SPICE installation.

```
unix% optimum -i SPITZER_S0_7130112_0002_6_A14567688_bksub.fits -o
SPITZER_S0_7130112_0002_6_A14567688_bksub.extract.tbl -b
SPITZER_S0_7130112_0002_6_A14567689_bkmsk.fits -pixel_scale 1.8 -fatal
29056 -nanDrop 1 -keepnans 1 -area_width 1001 -resamp 2 -w
cal/C15.0PRE25/b0_wavsamp.tbl -ridge
SPITZER_S0_7130112_0002_6_A14567688_bksub.ridge.tbl -p
cal/C15.0PRE25/b0_rectempl.fits -r
SPITZER_S0_7130112_0002_6_A14567688_bksub.rectified_flux.fits -r_unc
SPITZER_S0_7130112_0002_6_A14567688_bksub.rectified_unc.fits -r_bmask
SPITZER_S0_7130112_0002_6_A14567688_bksub.rectified_bmask.fits -
r_offset
SPITZER_S0_7130112_0002_6_A14567688_bksub.rectified_offset.fits -
clip_width 4.0 -clip_lambda0 6.0 -u
SPITZER_S0_7130112_0002_6_A14567690_bkunc.fits -height1 0 -norm 0
```

- *i* - Input 2D spectrum FITS file.
- *-o* - The name of the output extraction table file.
- *b* - The name of the mask FITS file that goes with the input 2D spectrum.
- *pixel_scale* - Pixel size (arcsec/pix) in input image.
- *fatal* - Fatal bit pattern for BMask. See Appendix B in the SPICE Manual for the allowed values and a description of how to calculate the fatal bit pattern. Default = 16384.
- *nanDrop* - Wavelength bins with NaNs will be removed from the output spectrum if this is set to 1. Default is 0, with the NaN fluxes set to -9999.
- *keepnans* - Set this to 1 if you wish a rectified pixel to be set to NaN if any of the corresponding pixels in the original image are NaNs. Default is to approximate with remaining non-NaN pixels.
- *area_width* - Width in pixels of rectified area (odd integer).

- resamp - Interpolation type to go from BCD image to rectified image. 0=nearest 1=weighted bilinear 2=unweighted bilinear
- w - The name of the **wavesamp.tbl* file. This file specifies the location of the spectral orders on the array in x-y coordinates. It consists of pseudorectangles, which describe the fractional pixels that comprise each wavelength in the spectrum. You should choose the appropriate file in the *cal/* subdirectory of your SPICE installation. The file you choose should match both the Spitzer pipeline version with which your data were processed and the IRS module with which you are working.
- ridge - The name of the output table from running the RIDGE module.
- p - Input stellar profile template FITS file.
- r - The name of the output rectified flux FITS file.
- r_unc - The name of the output rectified uncertainty FITS file. This is an optional intermediate product. SPICE will not write out this file unless an output filename is specified by the user.
- r_bmask - The name of the output rectified BMask FITS file. This is an optional intermediate product. SPICE will not write out this file unless an output filename is specified by the user.
- r_offset - The name of the output rectified offsets FITS file. This is an optional intermediate product. SPICE will not write out this file unless an output filename is specified by the user.
- clip_width - Width (in pixels of the original non-rectified image) of the extraction aperture at the reference wavelength. Default is half the width of the order.
- clip_lambda0 - Reference wavelength (in microns) of the aperture width specified with the "-clip_width" flag. Default is to use a constant width extraction aperture.
- u - The name of the uncertainty file that goes with the input 2D spectrum. If you do not specify an uncertainty file, the extraction will be profile-weighted only, and will use the default uncertainty value set by the flag "default_unc" for all pixels.
- height1 - Sets the wavesamp height. Set to 1 to force wavsamp height to 1 pixel; set to 0 to use actual height. Default = 0.
- norm - Normalize the fluxes. Default is no normalization.
 - 0 = no normalization
 - 1 = divide by height of wavesamp rectangle
 - 2 = divide by area of wavsamp rectangle

Set to 1 for S18.7 SL, LL, LH.
Set to 0 for S18.7 SH.
Set to 1 for S17.2 LH.
Set to 0 for S17.2 SL, LL, SH.
Set to 0 for all other CAL_SET versions.

The relevant output from the optimal extraction is

SPITZER_S0_7130112_0002_6_A14567688_bksub.extract.tbl, the one-dimensional optimally extracted spectrum, in units of electrons per second. Note that the regularly-extracted spectrum and the optimally-extracted spectrum have the same name in this example (and that is the default used by the GUI as well).

12.6 Run the TUNE Module

The final step is to use the TUNE module to convert the extracted spectrum from units of electrons per second to flux density (Jy). The conversion is only valid for the standard extraction width.

To run the TUNE module from the command line, try the following. Please note that you will have to modify the command below to provide the full path to the *cal/* directory in your SPICE installation.

```
unix% irs_tune -i
SPITZER_S0_7130112_0002_6_A14567688_bksub.extract.tbl -t
cal/C15.0PRE25/b0_fluxcon.tbl -m tune_down -o
SPITZER_S0_7130112_0002_6_A14567688_bksub.spect.tbl -a 3
```

You are free to modify the values for the above input flags. You can type 'irs_tune' with no argument at the prompt for a brief description of each possible input parameter. Also, you may wish to inspect the contents of the file *spice/cdf/*/*_settings.prop* to help you decide which values to give the input flags for data with a particular CAL_SET. The flags used above are described below.

- *i* - The output table from running the EXTRACT module in the previous step.
- *t* - Name of tuning table.
- *m* - mode (tune_up OR tune_down). Virtually all users will wish to use the "tune_down" option. The "tune_down" value means to apply the fluxcon table in the normal way, dividing by the factors. The "tune_up" value means to multiply by the factors. If you run *irs_tune* with *tune_down* on untuned data, you get tuned data. If you run *irs_tune* with *tune_up* on tuned data, you get the untuned data back.
- - The desired name of the output flux-calibrated spectrum, in units of Jy.

- a - Tuning components from the **_fluxcon.tbl* to apply. Default = 3. Below, "fluxcon" refers to the conversion factor from electrons per second to Jy, while "tune" refers to the remaining wavelength dependent calibration factors. Nearly every user will wish to set this value to 3 so that both the fluxcon and tune options are applied.

1 = fluxcon

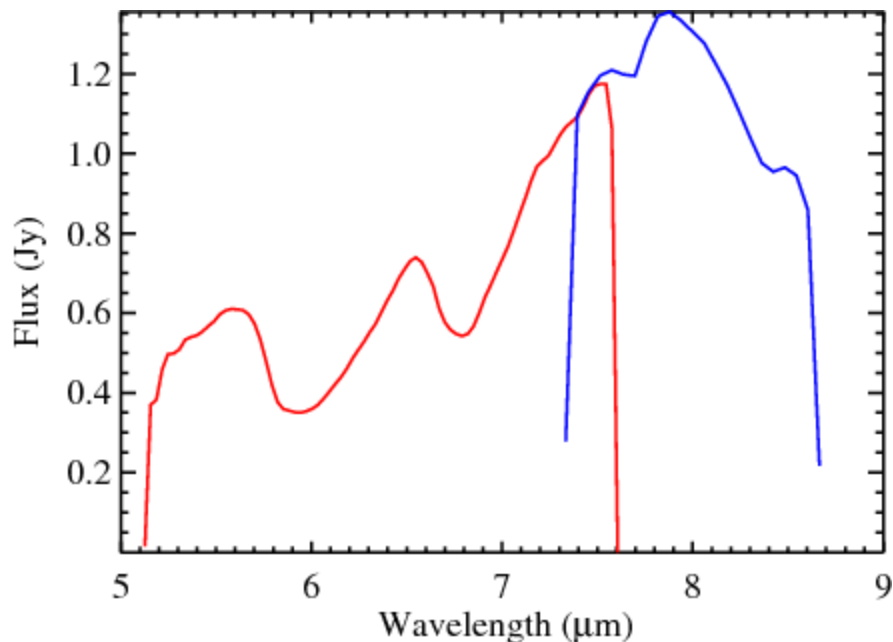
2 = tune

3 = both

The relevant output from issuing this command is

SPITZER_S0_7130112_0002_6_A14567688_bksub.spect.tbl

(http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/SPITZER_S0_7130112_0002_6_A14567688_bksub.spect.tbl), the flux-calibrated spectrum in units of Jy. (The linked spectrum is a tuned version of the regular extraction, not the optimal extraction). The spectrum is plotted below.



Recipe 13. IRSFRINGE: Remove fringes in data of HH 46/47

This is a demonstration of using IRSFRINGE to correct a spectral extraction from IRS data for the object Herbig-Haro 46/47 for fringing. We outline the basic steps for using IRSFRINGE as a generic example for correcting spectra of any point source data obtained with Spitzer IRS-Staring Mode. This specific example involves correcting a spectrum from the Long-High (LH) module, but it is generally applicable to any of the IRS modules which suffer from fringing. IRSFRINGE was developed and tested to run under **IDL 5.6**, and no guarantees are made of it running under other IDL versions.

13.1 Requirements

For this demo you will need the IRSFRINGE package

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/irsfringe/>, and BCD and post-BCD long-high IRS data from AOR 7130112. See Recipe 1 for instructions on downloading the data.

13.2 Extract the spectrum using SPICE

First, extract the spectrum of interest, in this case from the LH module, using SPICE. (See the SPICE demo for HH 46/47.) You will produce a **_spect.tbl* from the extraction. Or, use post-BCD pipeline output (**_tune.tbl*, **_spect.tbl*, or **_spect2.tbl* files). See Recipe 14 for details on using SPICE.

13.3 Start IRSFRINGE and load in data

To start IRSFRINGE type the following alias command in the running directory:

```
unix% irsfringe
```

Read in the IRS data product, either from the pipeline or from SPICE output. In this example, it is the *SPITZER_S3_7130112_8_1_E181777_coa2d.spect.tbl* in the *output/* directory, generated from a SPICE extraction. The spectrum is in IPAC table format and must be read into IRSFRINGE using the 'ipac2irs' command. The spectrum will have the name 'irs' in memory.

```
IDL>
```

```
irs=ipac2irs('/output/SPITZER_S3_7130112_8_1_E181777_coa2d.spect.tbl')
```

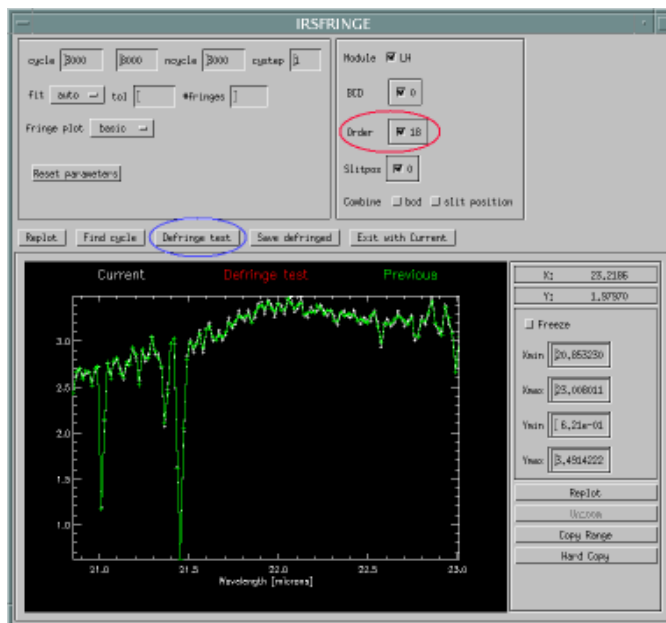
13.4 Run IRSFRINGE

To run IRSFRINGE on, e.g., order 18 of the spectrum, using the GUI, enter the following command:

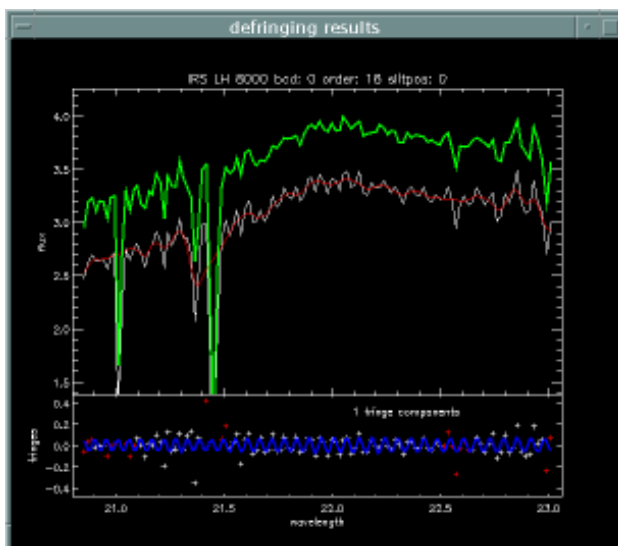
```
IDL> irsd=irsfringe(irs,order=18,/GUI)
```

The resulting defringed spectrum will have the name 'irsd' in memory.

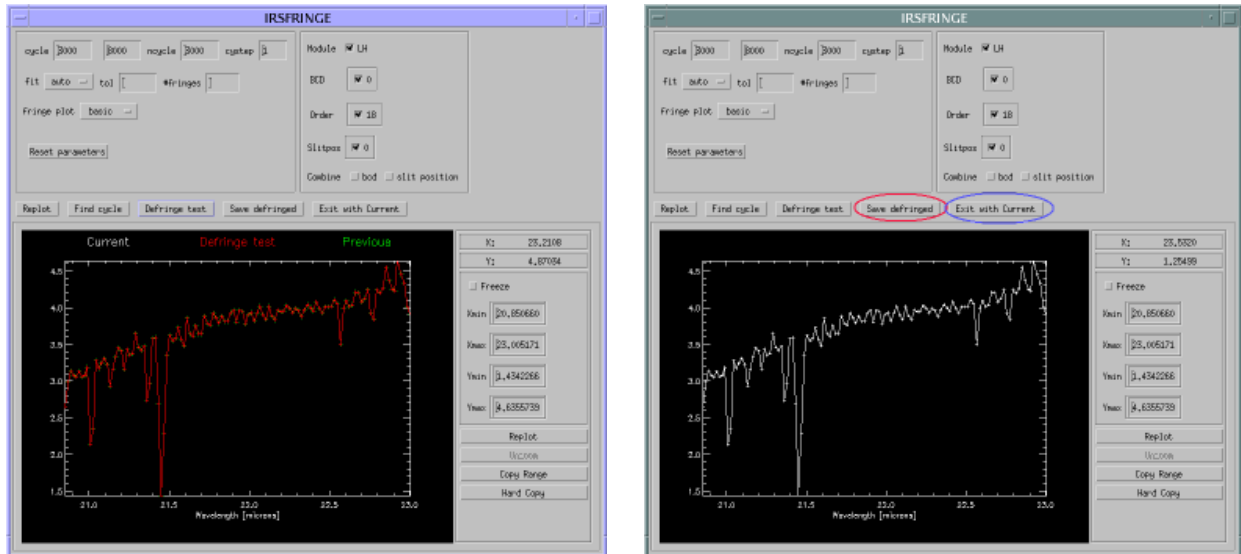
1. The GUI will have the appearance as in the figure below. We can see that IRSFRINGE will act on order 18 (circled in red). The green curve shows the original ('previous') spectrum and, since we have not yet acted on the spectrum, the updated ('current') spectrum, the white curve, is the same as the original. Pressing the 'Defringe test' button (circled in blue) will perform the defringing.



2. The following screen (see figure below) will pop up. The green line in the upper panel is the original ('previous') spectrum. The blue line in the lower panel is the best (sinusoidal) fit to the fringing pattern, represented by the white points, determined by the cycle finding routine in IRSFRINGE. The red line in the upper panel is the resulting defringing model. The white line in the upper panel is then the original spectrum corrected for the defringing model.



3. In the main GUI window (see figure below) will then appear the result of the defringe test, i.e., the original spectrum corrected for the defringing model. If satisfied with the test results, then press the 'Save defringed' button (circled in red in the figure below). The updated, defringed spectrum appears as the white line (the 'current' spectrum) in the window. Press the "Exit with Current" to exit the GUI with the saved defringed spectrum still in memory.



4. An output file of the defringed spectrum can then be written to local disk from the IDL prompt. In this example, we write the contents of memory, 'irsd,' to the output file *SPITZER_S3_7130112_8_1_E181777_coa2d.spect.defringed.tbl* in the directory *output/*. The defringed spectrum in memory needs to be written out once again to IPAC table format, hence, the command 'irs2ipac.' You're done! To exit IRSFRINGE, type 'exit' at the IDL prompt. All of the above, of course, can also be done on the IRSFRINGE command line, without using the GUI. See the IRSFRINGE documentation for command syntax.

```
IDL> irs2ipac, irsd,
/output/SPITZER_S3_7130112_8_1_E181777_coa2d.spect.defringed.tbl
IDL> exit
```

Recipe 14. SPICE: Spectrum Extraction of NGC4579

NGC4579 is an extragalactic source which was observed as part of the SINGS Legacy Science program. The source was mapped using both the low and high resolution modules of the IRS in Spectral Mapping mode. In this recipe, we outline the steps required to go from the Spitzer data archive to a basic spectrum using the spectral extraction tool SPICE. The illustration here is for only one position in the spectral map but can be generalized to any observation performed in the staring mode or spectral mapping mode.

14.1 Requirements

Install SPICE (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysistools/tools/spice/>), the Spitzer IRS Custom Extraction software and IRSFRINGE (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysistools/tools/irsfringe/>), the IRS Defringing software.

14.2 Download the example data set

Download the data associated with the SINGS observation of NGC 4579. Search by position or AOR ID to find AORs 9479424 (Sky Background for short-low data), 9494528 (Short-low module 1st and 2nd orders and Long-low module both), and 9499136 (Short-high and Long-high modules). The example below walks through the steps for AOR 9494528, 2nd order. For a better understanding of the wavelength range and names of the modules see the IRS pocketguide.

The delivered, unzipped data will be in a directory with the name corresponding to the AOR ID (*r9494528/*) with subdirectories such as *ch0/*, *ch1/*, *ch2/*, *ch3/* corresponding to the Short-low (short wavelength low resolution = SL), Short-high (SH), Long-low (LL) and Long-high (LH) modules respectively. In this particular example only the *ch0/* and *ch2/* subdirectories are created corresponding to the low resolution modules.

Each of the *ch*/* subdirectories has the **bcd.fits* files in the *bcd/* subdirectory. The **bcd.fits* files are the ones to start with to obtain a final reduced spectrum for the example data. You will also need the **func.fits* and **bmask.fits* files which are the uncertainty files and pixel status masks, respectively. You can edit the *bmask* files by hand to flag pixels which might be bad in your data. To identify which bits in the *bmask* file need to be changed to flag a pixel, read the SPICE online help within the GUI, in particular "Bmask Status Bits". Check the SSC web site to get campaign specific bad pixels masks (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/irs/calibrationfiles/>) which were created from the IRS darks.

14.3 Have a quick look at the data

Before attempting any data reduction, visualize your AOR in Spot to understand the orientation of the observations and the sequence in which the files were generated. The AOR used and the visualization of these particular spectral mapping observations are shown below. As can be seen, the short wavelength modules each consist of 18 steps in the perpendicular direction while the LL1 module consists of 6 steps parallel to the slit and 10 steps perpendicular to the slit.

Unique AOR Label:

Target: **NGC4579** Type: **Fixed Single**
 Position: **12h37m43.6s,+11d49m06s**

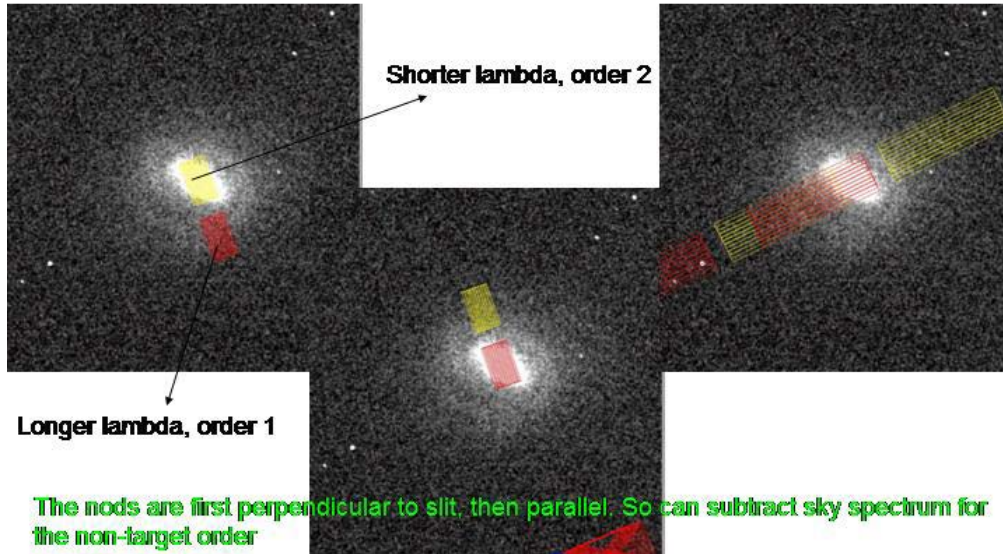
Peak-Up Settings
 Choose one Peak-Up mode

Peak-Up Disabled

Instrument Settings
 Number Of Map Cycles:

IRS Slit	On	Ramp Du...	Cycles	Pointings Para...	Step Par. (")	Pointings Perp...	Step Perp...
Hi 9.9-19.6 um	<input type="checkbox"/>	6	1	5	100.00	5	100.00
Hi 18.7-37.2 um	<input type="checkbox"/>	60	1	5	100.00	5	100.00
Low 5.2-8.7 um	<input checked="" type="checkbox"/>	14	1	1	28.35	18	1.85
Low 7.4-14.5 um	<input checked="" type="checkbox"/>	14	1	1	28.35	18	1.85
Low Short Both	<input type="checkbox"/>	6	1	5	100.00	5	100.00
Low 14.0-21.3 u...	<input type="checkbox"/>	14	1	5	100.00	5	100.00
Low 19.5-38.0 u...	<input type="checkbox"/>	14	1	5	100.00	5	100.00
Low Long Both	<input checked="" type="checkbox"/>	30	1	6	79.00	10	5.08

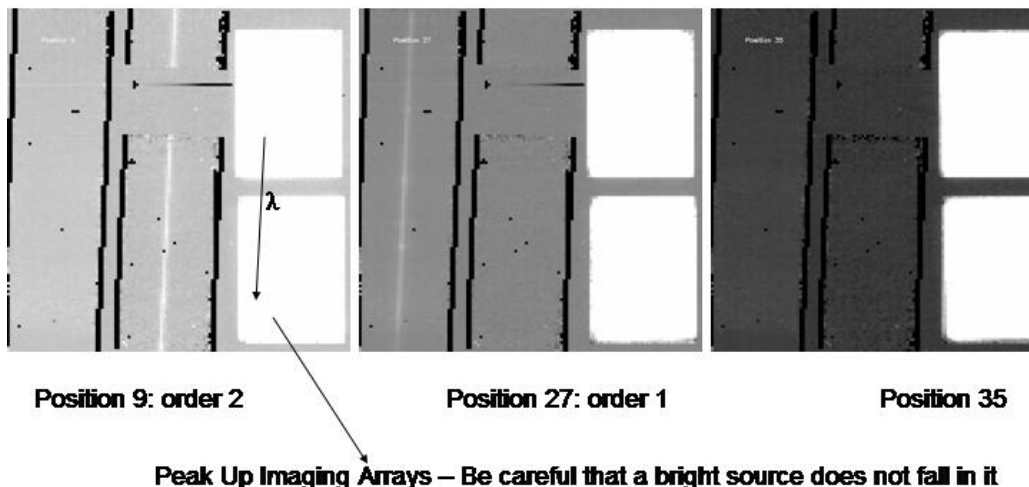
Always check if what you get is what you expected by comparison with SPOT e.g. Roll angle should be 294.9 on date of observation June 28, 2004. Comparison with the focal plane layout indicates this is the correct orientation. Note that the SL and LL slits are not parallel.



Now view your data in 2D format using your favorite fits file viewer such as DS9 or SAOIMAGE. The short-low data is shown on the left and the long-low data on the right for different positions in the spectral map. Note that in SL order 2, the nuclear source is in the slit in position 9. In SL order 1, the nuclear source is in the slit in position 27 for these observations.

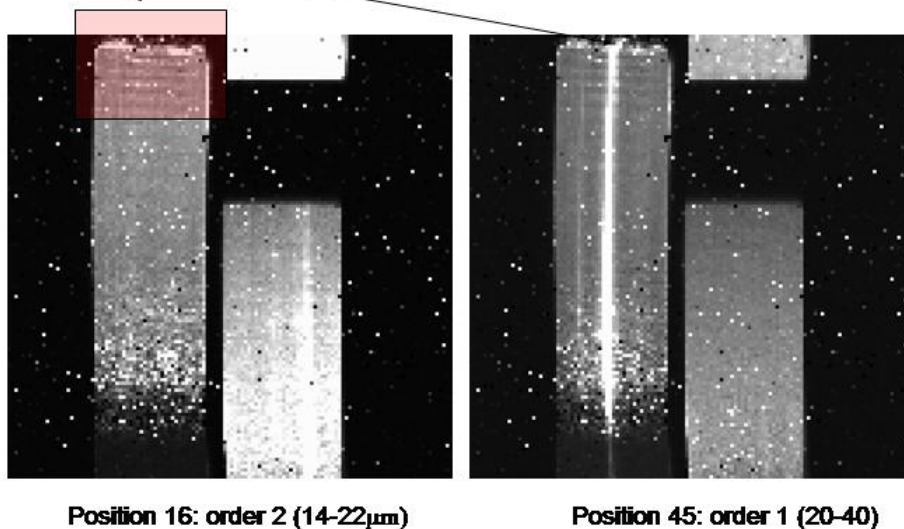
SL: What does the data look like ?

This particular AOR has 18 steps through the galaxy for each SL order, only the 9th position is on the nucleus.



LL Data

Fringing due to filter delamination. Alleviated by dividing by a flat but if still present, use IRSFRINGE software to remove. However, dividing by a flat for an extended source might make fringing worse.



14.4 Spectral extraction using SPICE

1. Launch SPICE. If SPICE is in your path, for example:

```
unix% spice &
```

2. Open a point source extraction flow in SPICE: **File --> Open Spice Generic Template --> Point Source with Regular Extract.**
3. Load the BCD, uncertainty and mask files into SPICE. This is done by clicking on the "Modify" button for the "Image File" in the **Initialize Parameters and Files** module. If you are using archive file names, the corresponding mask and uncertainty files will be selected automatically. The BCD image will be displayed in the FITS window.

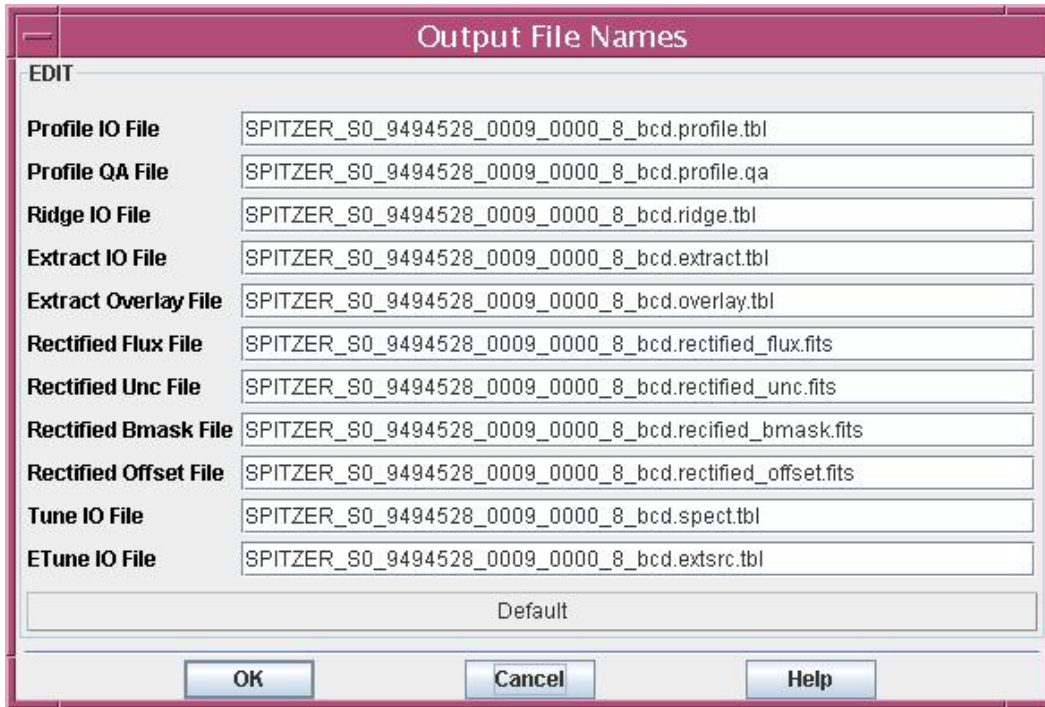


You can change the viewing parameters such as the stretch (using the controls on the right of the FITS window) or zoom of the display (controls on the left of the full SPICE window).



4. Check the name of the output files. Each time a module is run, the output files are overwritten, so you want to be careful in changing the output filename when you change the extraction parameters. First,

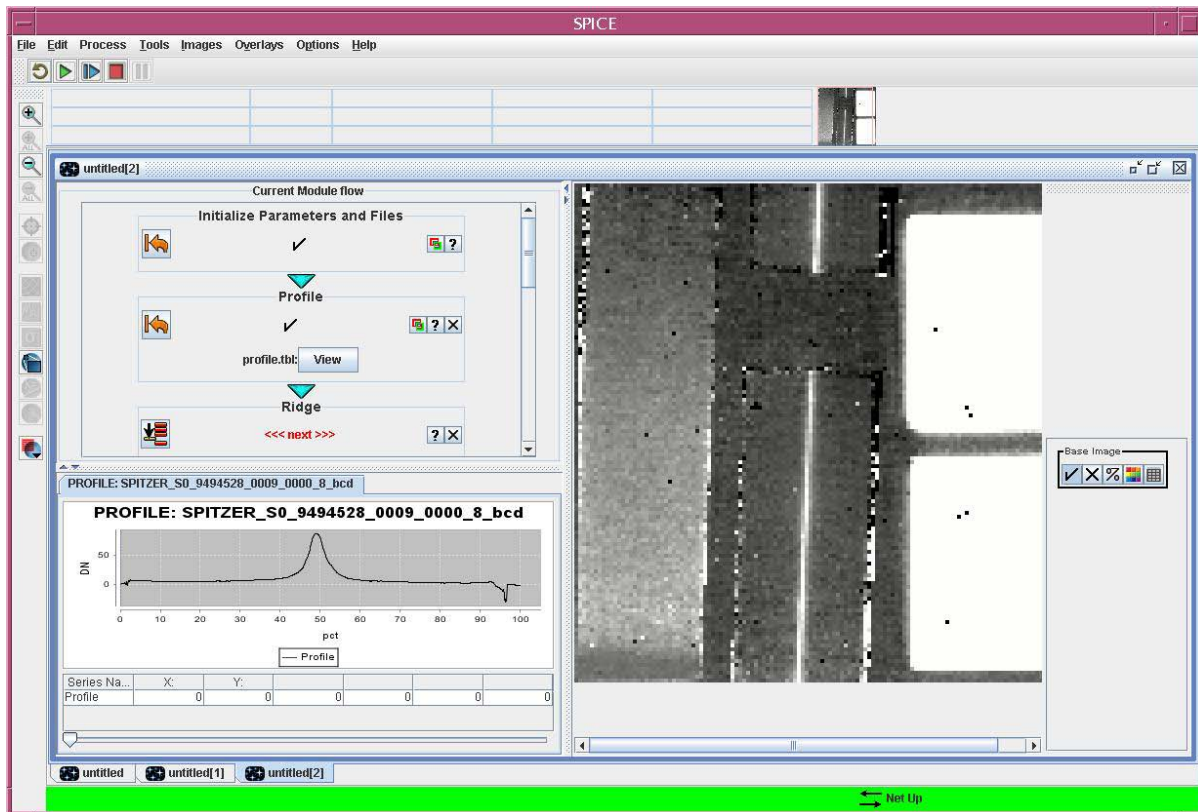
select your output directory using the "Modify" button for "Current Output Directory". Next, you can review and modify the output file names using the "EDIT" button for "Output Files".



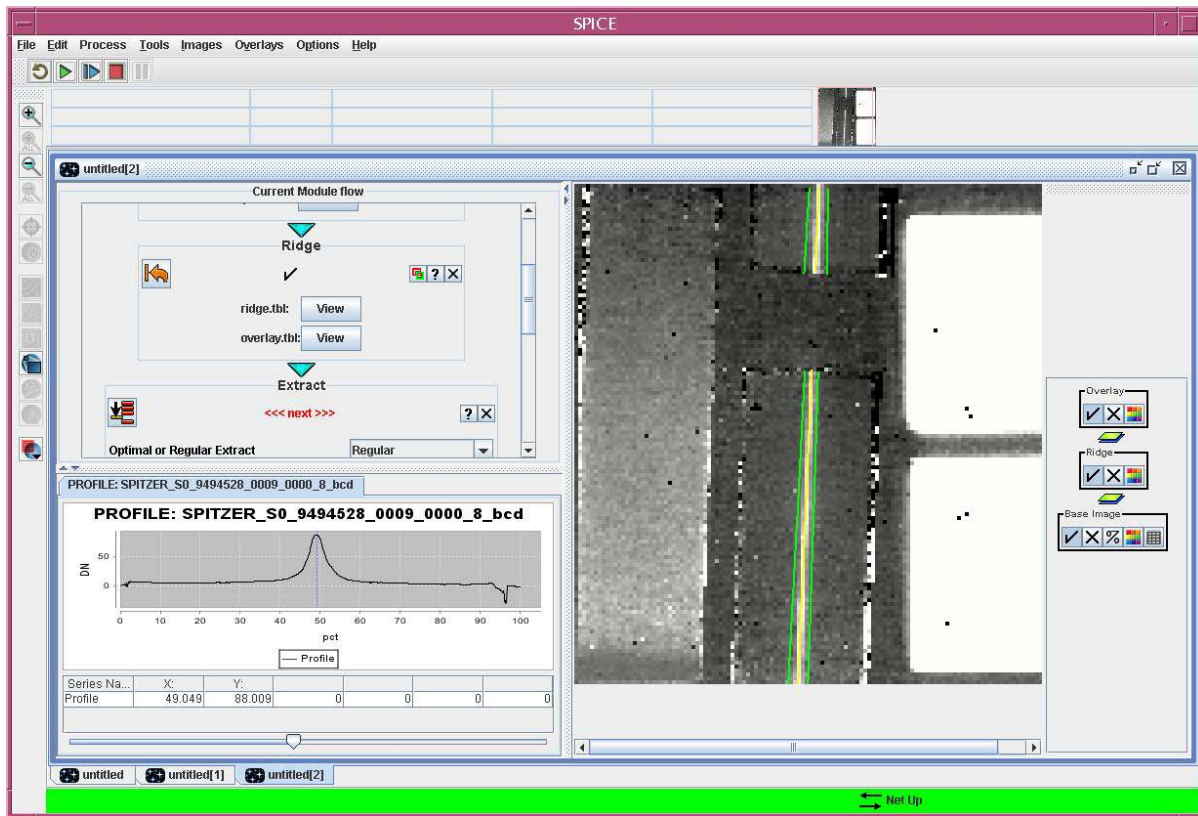
5. Run the **Initialize Parameters and Files** module. Click the "run" button in the upper left of the module.



6. Run the **Profile** module. In this case, you do not need to change the Low Res Order of the profile. If there were an object in the non-target order (SL-1 in this case) you could change the order of the extraction. Again, run the module by clicking the "Run" button. The profile module collapses the spectrum in wavelength space for a particular slit order. This maximizes the S/N to help determine where the source is within the slit. When the module is run, the spatial profile will be displayed in the Plot window.



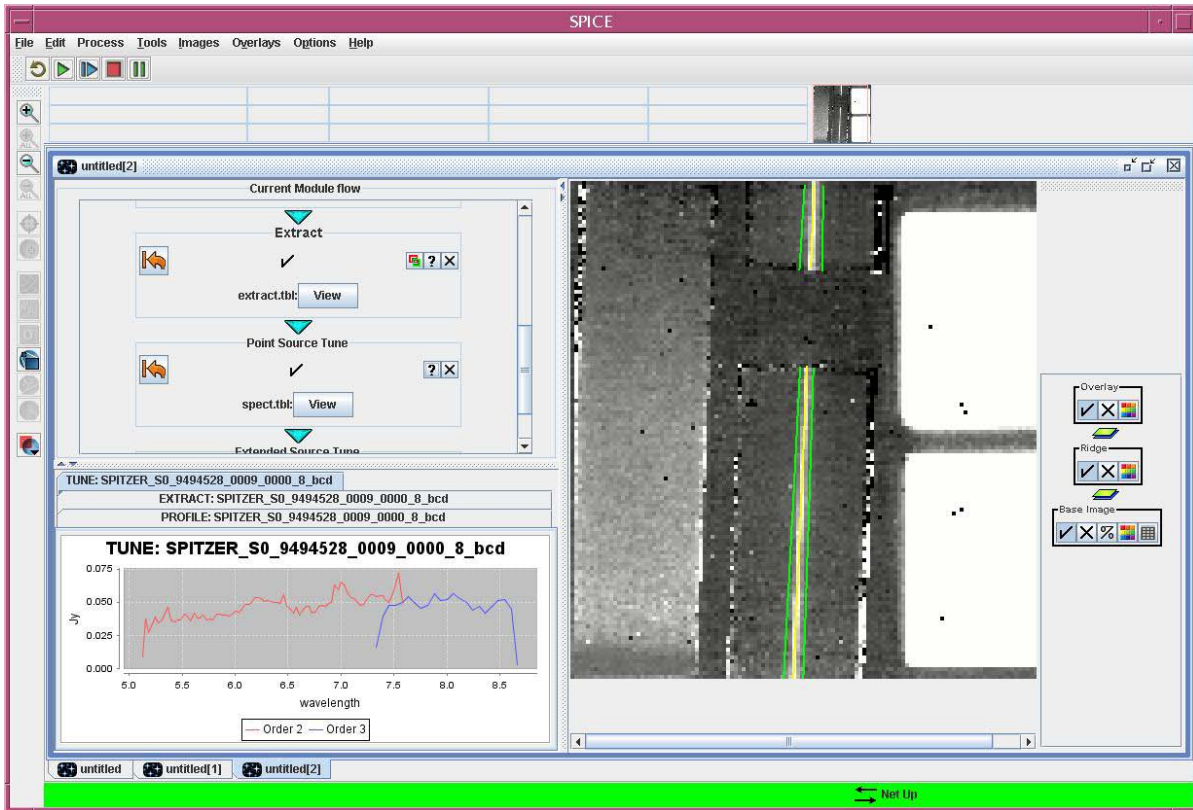
7. Run the **Ridge** module. If the "Percent Setting" is set to "Default" then the peak in the profile is automatically measured from the spatial profile by SPICE. This works well for bright, isolated, compact sources like the one in this example. Alternately, use the "Manual" option to enter a number as a percent of the slit width where the source should be extracted from. This is particularly useful when you want to extract the spectrum of the sky or the spectrum of a secondary source which is not the brightest source in the slit. The percentage appears as a dotted-blue line in the plot. Moving the line with the slider changes the percentage in the module, but moving the line by clicking on the plot does not. When you run the module, the trace and extraction width will be overlaid on the FITS image.



- Specify the width over which the spectrum should be extracted. This can be re-defined in the next module, **Extract**. By default, an observation in SL-2 will be extracted as a point source with a narrow window which expands with wavelength. This default option is fine for the current example. Alternately, you can manually specify the width in pixels for a particular wavelength. The width of the extraction window increases with increasing wavelength to factor in the diffraction-limited point spread function. If you specify a width at 0 microns, then a constant width extraction window is used which is independent of wavelength. The overlay in the FITS window show which part of the slit is being extracted. In this module, there are also other options such as extended source extraction "ExtSrc", masking of specific bits and interpolation of NaNs which can help you improve the fidelity of your spectrum.

Run the module by clicking on the run button. The extracted spectrum will be displayed in the plot window in units of electrons per second.

- Flux calibrate the extracted spectrum using **Point Source Tune**. Click the run button for this module (there are no parameters) to obtain a flux calibrated spectrum for a point source. If you use the default options in each step of the process, the output of tune should be exactly the same as that of the post-BCD files that you downloaded from the Spitzer archive. The output of the module is a file called **spect.tbl* unless you have changed the output file name.



10. Remove fringes using IRSFRINGE. If you are working with LL data or high resolution data, it is likely that you will have residual fringes. This can be removed using the IDL procedure IRSFRINGE. Fire up IDL and at the IDL prompt:

```
IDL% irs=ipac2irs('spect.tbl')
IDL% irsd=irsfringe(irs,order=1)
```

IRSFRINGE has a number of different options which are discussed in the IRSFRINGE User's Guide. These allow you to defringe only particular orders or a particular wavelength range or mask particular spectral features that might affect the defringing. Defringing is generally an empirical tool which should be used at the observer's discretion since it is not always clear how many sine waves need to be fit to the data or what their relative frequencies are.

14.5 Miscellaneous Notes

SPICE will use calibration files which are consistent with the version of the pipeline that was used to process your data (see the CREATOR keyword in the header of your *bcd.fits* files). If they were processed with S11.0.2, SPICE will generate an error if you do not have the calibration files that correspond to S11.0.2.

SPICE has a batch mode which allows you to process large numbers of files in exactly the same way once you have identified the appropriate extraction parameters for your data. The usage is similar to the above example. Open the Batch flow using **File --> Open Batch Spice Generic Template**.

The default options for the steps outlined in this illustration i.e. profile, ridge, extract, tune and defringe will result in an output spectrum which is similar to the Spitzer IRS pipeline. You will want to improve the quality of your data before undertaking these steps by performing:

1. Sky subtraction: The pipeline post-BCD sky subtracted product is just one nod position from the other. You may want to create a supersky by taking a median of the BCD files from the off source position and subtract it from the on source data.

2. Creating bad/hot/rogue pixel masks: You may need to make new masks which are appropriate for your data by flagging bits in the bmask files.

3. Optimal Extraction: Improved S/N can be obtained for point sources using Optimal extract. This option runs a separate extraction module, which weights the extraction by the point source profile. Optimal extraction is documented in the SPICE GUI internal help pages. Open a separate optimal extraction flow from the file menu.

4. See the IRS Instrument Handbook which describes the steps you should undertake before publishing your IRS data.

Recipe 15. SPICE: Spectrum Extraction of HH 46/47

This is a recipe using SPICE to extract a spectrum from IRS data for the object Herbig-Haro 46/47. We outline the basic steps for using SPICE as a generic example for handling any point source data obtained with Spitzer IRS-Staring Mode. This specific example involves an extraction from the SL2 module, but it is generally applicable to all low-resolution IRS data. This demonstrates only a point source extraction; SPICE is currently capable of extracting spectra for extended sources. For a discussion, see the SPICE Manual on our website.

15.1 Requirements

- You will need the Spitzer archive tool to download the example data set.
- You will need SPICE (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/spice/>), the Spitzer IRS Custom Extraction software, to extract your spectrum.

15.2 Download and examine the example data set

A recipe for downloading data can be found in Recipe 1. For this example we will use the IRS data for the Herbig Haro object HH 46/47, AOR 7130112. It is sufficient to download the BCD data for this demonstration.

In this recipe, we will focus on SL2 (short-low, order 2) data. Therefore, once the data are downloaded, you will be most interested in the **bcd.fits* files in the subdirectory called *r7130112/ch0/bcd/*. (All the data in *ch0/* is from the short-low module, while *ch1/* holds short-high data and *ch3/* holds long-high data.)

```
unix% ls *bcd.fits
```

```
SPITZER_S0_7130112_0000_0000_6_bcd.fits
SPITZER_S0_7130112_0000_0001_6_bcd.fits
SPITZER_S0_7130112_0000_0002_6_bcd.fits
SPITZER_S0_7130112_0000_0003_6_bcd.fits
SPITZER_S0_7130112_0001_0000_6_bcd.fits
SPITZER_S0_7130112_0001_0001_6_bcd.fits
SPITZER_S0_7130112_0001_0002_6_bcd.fits
SPITZER_S0_7130112_0001_0003_6_bcd.fits
SPITZER_S0_7130112_0002_0000_6_bcd.fits
SPITZER_S0_7130112_0002_0001_6_bcd.fits
SPITZER_S0_7130112_0003_0000_6_bcd.fits
SPITZER_S0_7130112_0003_0001_6_bcd.fits
```

```
SPITZER_S0_7130112_0004_0000_6_bcd.fits
SPITZER_S0_7130112_0004_0001_6_bcd.fits
SPITZER_S0_7130112_0005_0000_6_bcd.fits
SPITZER_S0_7130112_0005_0001_6_bcd.fits
```

To identify each BCD file, examine the "FOVNAME" keyword in the fits headers. An easy way to do this is to use the `imhead` command available as part of the WCSTools package (<http://tdc-www.harvard.edu/wcstools/>), but you can use your own favorite method.

```
unix% imhead *bcd.fits | grep FOVNAME
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Center' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Center' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Center' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Center' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Sweet_Spot' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Sweet_Spot' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Sweet_Spot' / Field of View Name
FOVNAME = 'IRS_Blue_Peak-Up_FOV_Sweet_Spot' / Field of View Name
FOVNAME = 'IRS_Short-Lo_2nd_Order_1st_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_2nd_Order_1st_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_2nd_Order_2nd_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_2nd_Order_2nd_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_1st_Order_1st_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_1st_Order_1st_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_1st_Order_2nd_Position' / Field of View Name
FOVNAME = 'IRS_Short-Lo_1st_Order_2nd_Position' / Field of View Name
```

You can see that the first eight files are associated with peak-up acquisition, and only the last eight files contain spectroscopic data. As stated above, we are interested in reducing the SL2 data. As part of the normal observations sequence, the target is placed in each of two positions along the SL2 slit. The general procedure is to extract a 1D spectrum for each nod position, and then average the nods. In the following, we will focus on extracting a 1D spectrum from the 2nd nod position. You can follow an analogous procedure to extract a 1D spectrum from the 1st nod position. In the last step, we'll combine the two nods.

15.3 Subtract the background

The next step is to use the downloaded BCDs to create a background-subtracted 2D spectral image on which to perform the extraction. There are two steps to this: building the background and subtracting it. SPICE does not perform either of these steps for you. You will have to use an image processing program, such as IRAF or IDL. When doing so, be sure to retain the FITS header from your BCD image. The basic steps are outlined below.

15.3.1 Building the background

You can build your background for SL2 nod 2 from the other nod position, from the other order, or both.

1. **Build your background from the other nod position.** As part of the default observing sequence, staring mode targets are placed in each of two positions along the slit. When the source is in position 1, the sky will be in position 2. Therefore, if we want to build a sky image for position 2, we can use the position 1 frames. In our example, this would mean building sky frames from sources with FOVNAME header values of 'IRS_Short-Lo_2nd_Order_1st_Position'. Thus, our background frames would be:

```
SPITZER_S0_7130112_0002_0000_6_bcd.fits
SPITZER_S0_7130112_0002_0001_6_bcd.fits
```

Using IRAF or IDL, you can average these frames to create a background image. In this case, there are only two images, which is not ideal for removing bad pixels.

2. **Build your background from the other spectral order.** In this example, both SL1 and SL2 observations were taken. When a target is placed in the SL1 slit, the sky is in the SL2 slit. Therefore, we can use SL1 observations to build up a sky image for SL2 data. In our example, the SL1 images are:

```
SPITZER_S0_7130112_0004_0000_6_bcd.fits
SPITZER_S0_7130112_0004_0001_6_bcd.fits
SPITZER_S0_7130112_0005_0000_6_bcd.fits
SPITZER_S0_7130112_0005_0001_6_bcd.fits
```

Using IRAF or IDL, you can average these frames to create a background image. In this case, we have four images to combine, which will result in a sky image with lower noise than the previous option.

3. **Build your background using both the other nod position and the other spectral order.** To build a background frame with even higher signal-to-noise, we can use both the SL2 images taken in Position 1 and the SL1 images:

```
SPITZER_S0_7130112_0002_0000_6_bcd.fits
SPITZER_S0_7130112_0002_0001_6_bcd.fits
SPITZER_S0_7130112_0004_0000_6_bcd.fits
SPITZER_S0_7130112_0004_0001_6_bcd.fits
SPITZER_S0_7130112_0005_0000_6_bcd.fits
SPITZER_S0_7130112_0005_0001_6_bcd.fits
```

This gives us six images with which to build the sky frame, and therefore better statistics than the previous two methods. This is the option we choose for this recipe. One of the two options above may be sufficient for your needs depending on the number of frames available.

15.3.2 *Subtracting the background*

The next step is to create a single, background-subtracted 2D image of the SL2 nod 2 spectrum. There are two strategies you can employ:

1. **Stack the background-subtracted BCDs.** Start by background-subtracting each of the SL2-nod2 BCDs:

```
SPITZER_S0_7130112_0003_0000_6_bcd.fits  
SPITZER_S0_7130112_0003_0001_6_bcd.fits
```

Make sure you retain the header information from the BCDs (and not from the background image you created), so that SPICE knows that this is a SL2-nod2 spectrum.

Now you have two background-subtracted BCDs. You can combine these using a simple average to create a single, 2D background-subtracted image of the SL2-nod2 spectrum.

2. **Stack the BCDs, then subtract the background.** Alternatively, you can average together the SL2-nod2 BCDs and then subtract the background from the result. Again, you must make sure that the result carries the header information from the SL2-nod2 BCDs, and not from the background image. This should provide a very similar result to the previous method.

Either way you choose, be sure to create new **unc.fits* and **bmask.fits* files as well. You should propagate the pixel-by-pixel uncertainties and combine the mask bit flags using the "OR" operator.

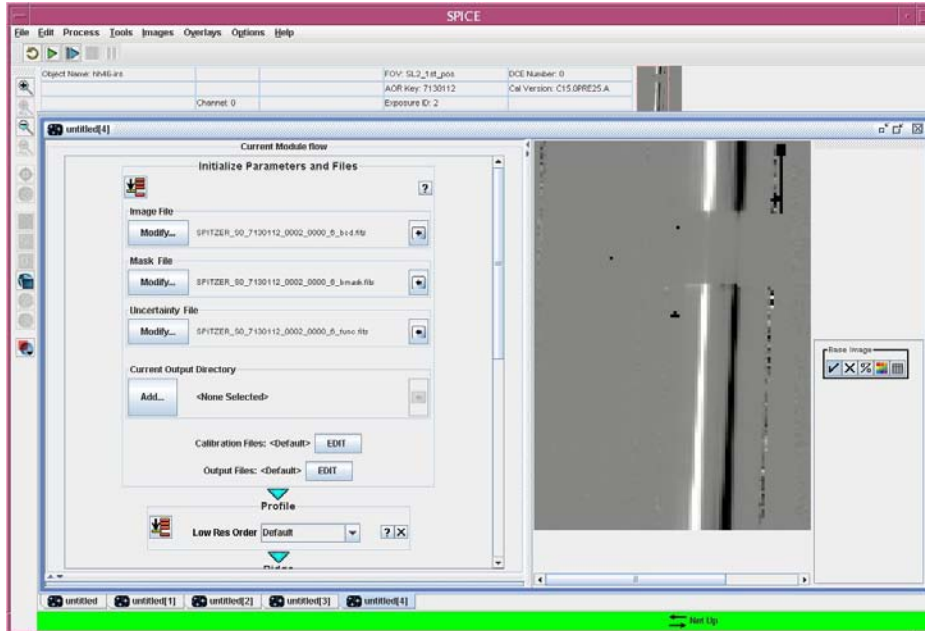
15.4 Choose a SPICE extraction template

1. Launch SPICE.
2. Within the SPICE GUI, open a point source extraction flow using **File --> Open Spice Generic Template --> Point Source with Regular Extract.**

15.5 Run the Initialize Module

1. Load the bcd, uncertainty and mask files into SPICE. This is done by clicking on the "Modify" button for the "Image File" in the **Initialize Parameters and Files** module. If you are using archive file names,

the corresponding mask and uncertainty files will be selected automatically. The BCD image will be displayed in the FITS window, as shown in the image below.

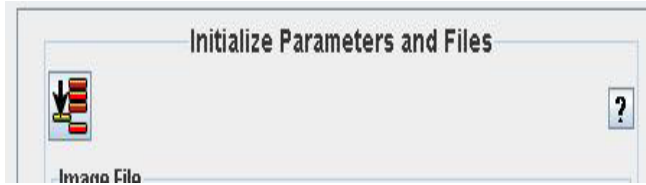


You can change the contrast ("stretch") of the 2-D image on the right by clicking on the color-grid button at the right of the FITS window. You can zoom the image using the controls to the left of the full SPICE window. See figure below -- the controls are circled in red.



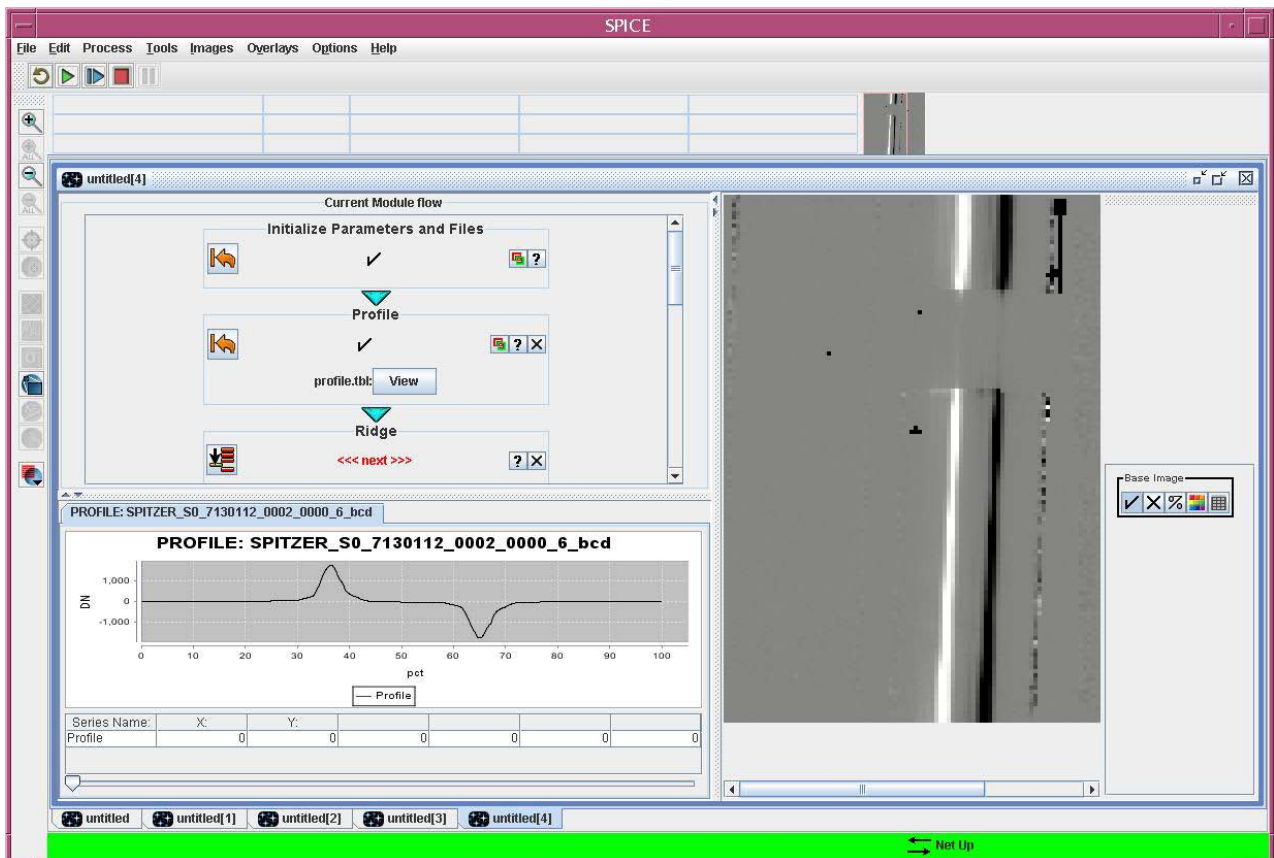
2. Select the output directory for the output **.profile.tbl*, **.ridge.tbl*, **.extract.tbl*, and **.spect.tbl* files generated by the extraction. The ultimate output table of interest is the **spect.tbl* -- this is your extracted 1-D spectrum in (ascii) IPAC table format. You can make the selections by pressing the "Output" button.

3. When you have set all the input parameters, run the **Initialize Parameters and Files** module, using the run button in the upper left of the module window:



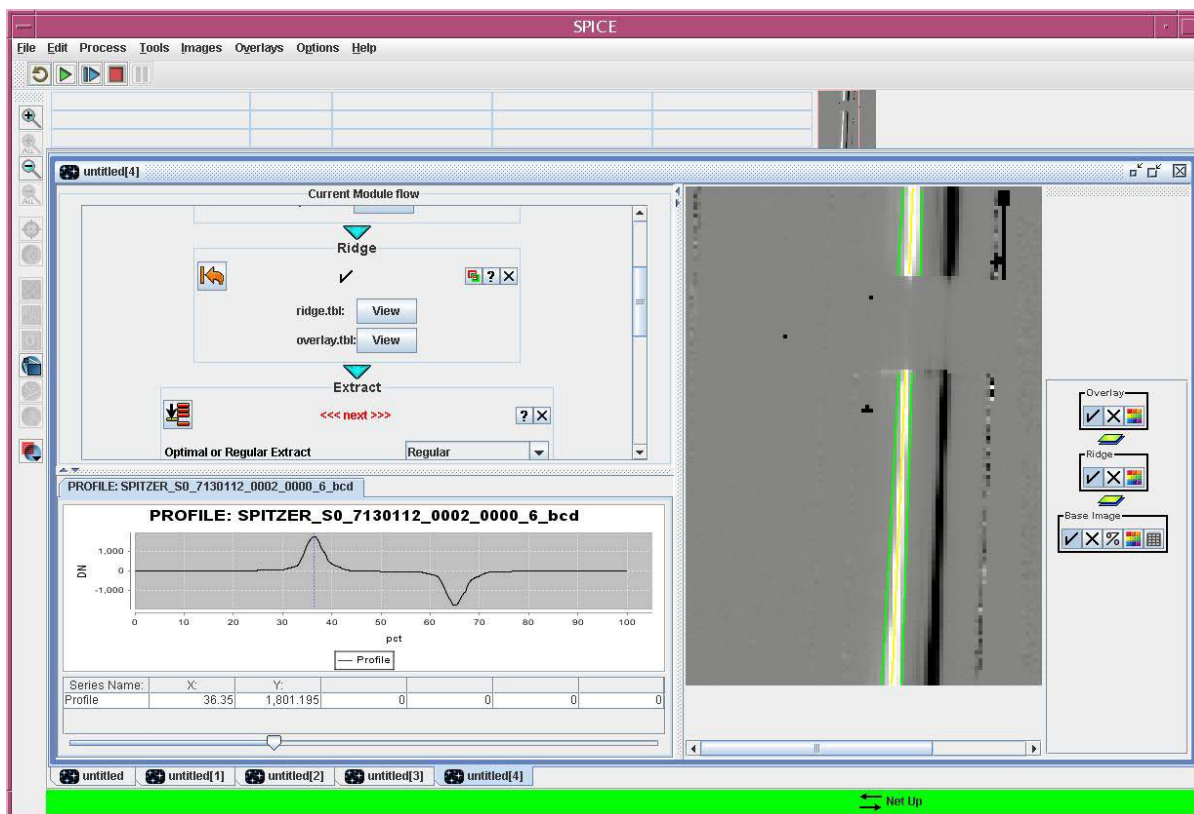
15.6 Run the profile module

To begin the extraction, establish the wavelength-collapsed average profile in the spatial direction across the 2-D background-subtracted spectrum, using the **Profile** module. See the figure below. Note that for the background-subtracted spectrum, a "positive" profile (your spectrum) and a "negative" profile (the spectrum in the other nod), relative to a zero level, are seen as the output.



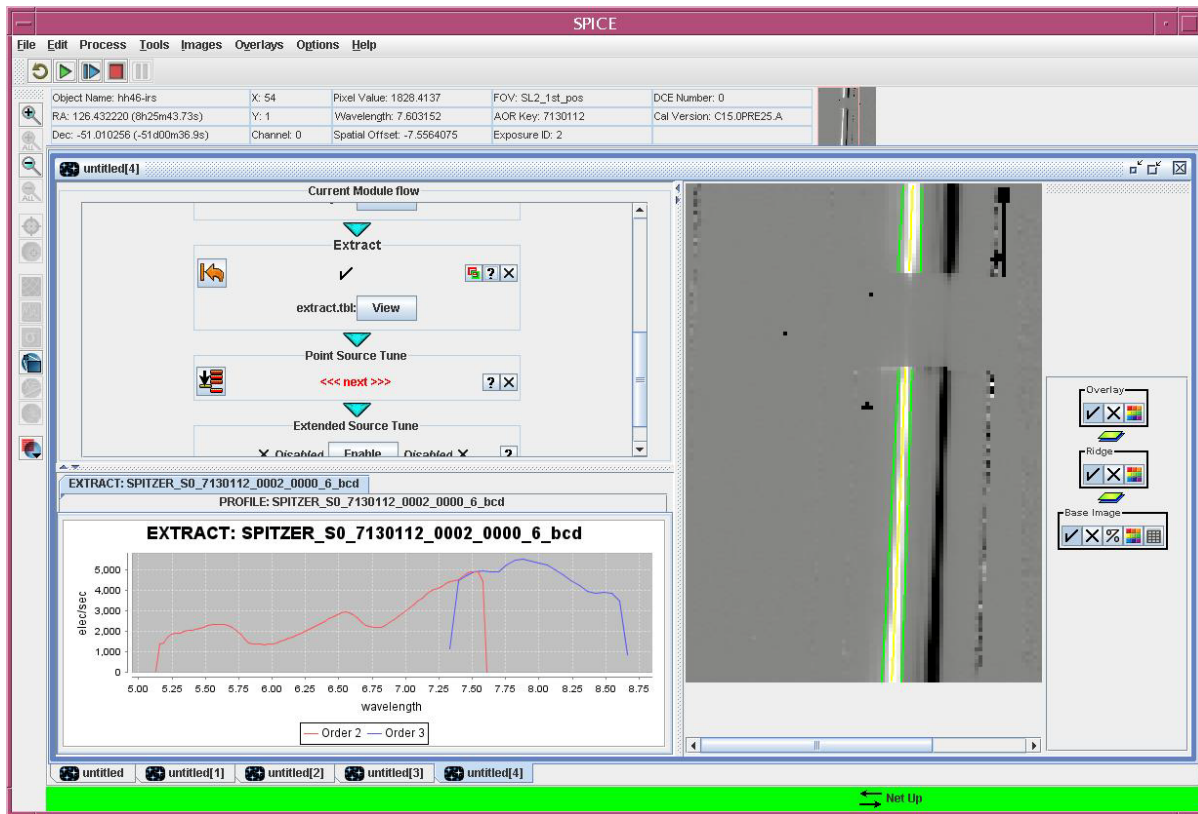
15.7 Run the Ridge module

Next, establish the (peak) ridgeline of the spectrum in the dispersion direction along the 2-D spectrum. See bottom figure below. Running the **Ridge** module will perform the function. You can either allow SPICE to automatically derive the ridge peak or set the ridgeline peak manually.



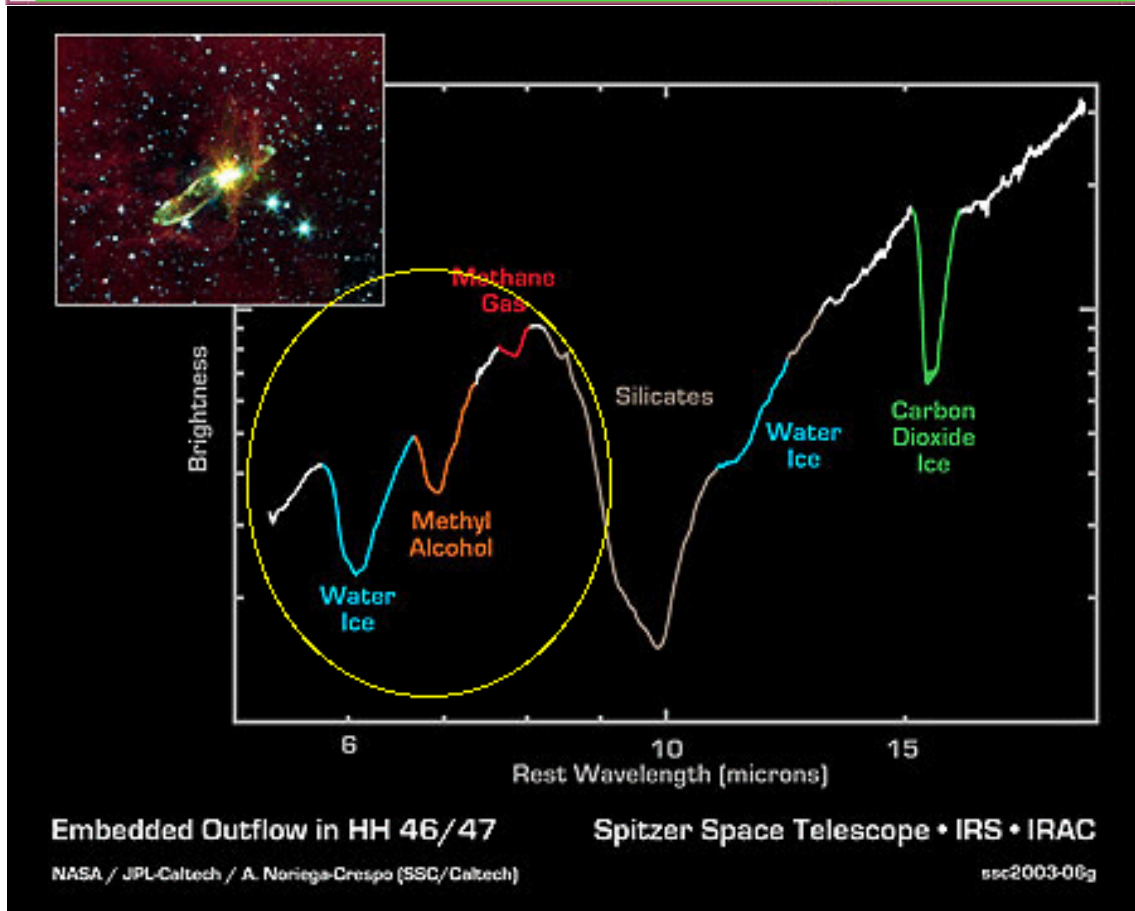
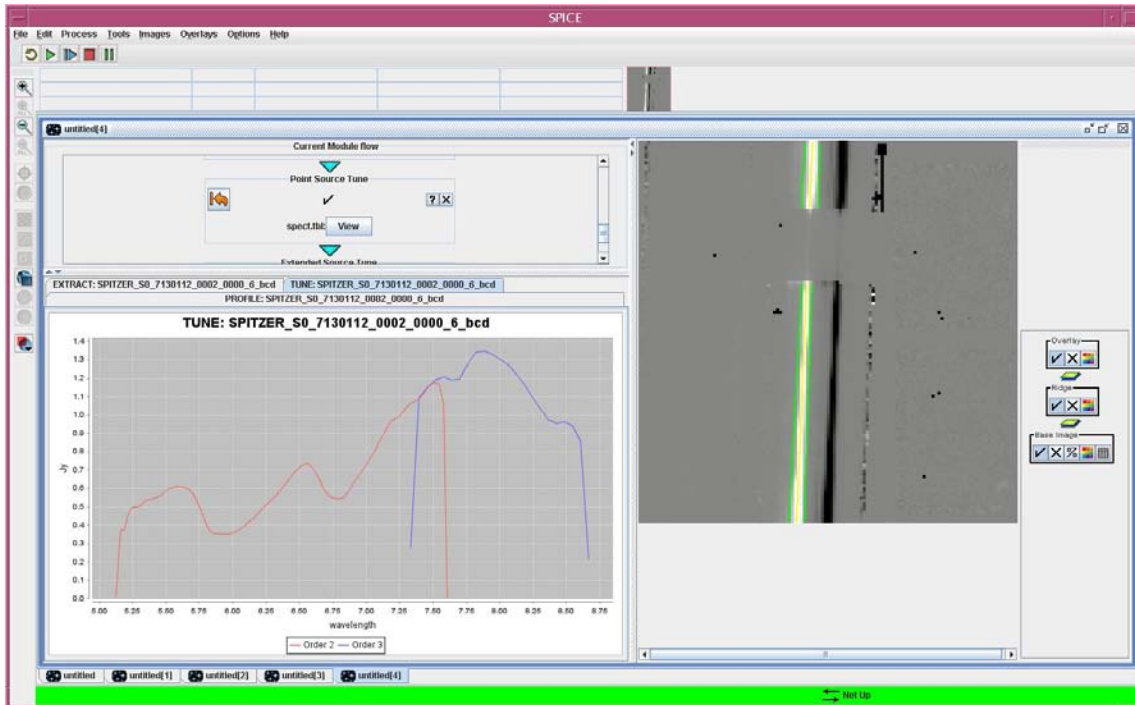
15.8 Run the Extract module

To extract the spectrum, run the **Extract** module, which is next in the flow. The extraction can be done with the default parameters for this example. For the low-resolution modules, the spectrum is extracted along the Ridge location, in accordance with the wavelength-dependent Point Spread Function (PSF) and the spectral trace (with pipeline-optimized width). The Extract function can employ a window with a different width ("Manual" width), but the width will still scale with wavelength, unless a full-slit extraction ("ExtSrc" width) is specified. Note that the output of the extraction is still in instrumental units, i.e., electrons/sec.



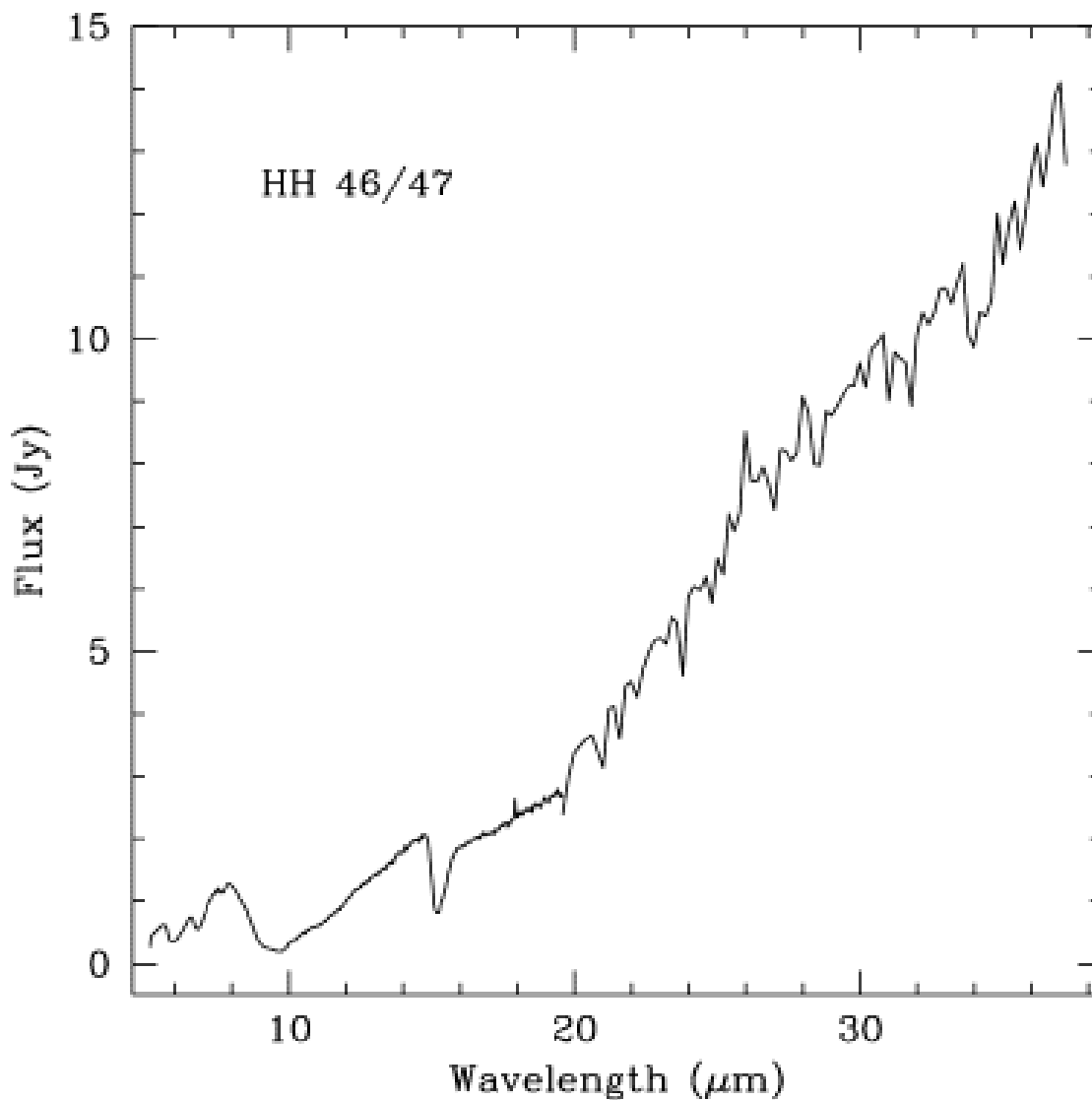
15.9 Run the Tune module

You still need to "tune" the extraction by applying the flux conversion from instrumental to absolute flux units. To do this, run the **Point Source Tune** module. This function will correct the slope and curvature of each order by applying the polynomial coefficients in the `*_fluxcon.tbl` file. This correction is based on an order-by-order comparison of calibration data to standard star model spectra. The flux units are now in Janskys (Jy). This completes the spectral extraction for this module. Note that the so-called "bonus order" of SL2 is somewhat mismatched in flux. You can choose to keep this bonus order, or exclude it (since it overlaps with an order in SL1). Our extraction looks remarkably like the short-wavelength portion of the one in the Spitzer press release and the published paper (Noriega-Crespo et al. 2004, ApJS, 154, 352; see bottom figure below, circled *in yellow*)... as it should!



15.10 Combine Extractions and Examine Results

You can perform custom extractions on all the modules and orders for these observations, merge the various *spect.tbl* tables together (outside of SPICE!), and obtain a spectrum similar to what is shown below.



For reference, we provide our extracted spectrum as an ASCII table on our SSC web site (http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/hh46_spect.dat). Your extraction obtained from these data may differ slightly.

Recipe 16. SPICE: Spectrum Extraction for Comet S-W 1

This demonstration uses SPICE to extract a spectrum from IRS data of the Comet Schwassmann-Wachmann 1. We describe the steps required to background-subtract your data and use the SPICE tool to perform a custom extraction of your target's spectrum. Background subtraction is particularly important for targets on a high background, such as that found in the ecliptic plane. Similarly for a moving target such as a comet, a custom extraction, rather than the standard pipeline product, may be important to separate the effects of nucleus and coma, and to otherwise correctly extract a slightly extended object, or one with an unusual morphology. This demonstration then, serves as a generic example for handling moving target data obtained with Spitzer IRS-Staring Mode. This specific example involves an extraction from the LL2 module, but it is generally applicable to all of the IRS modules.

16.1 Requirements

1. Download and install SPICE (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/spice/>).
2. Download the BCD and post-BCD data associated with the observation of Comet Schwassmann-Wachmann-1 in AOR 6068992. See Recipe 1 for instructions on how to download the data.

16.2 Step-by-Step Guide:

1. Perform the background subtraction. For IRS staring data in LL2, for point sources and slightly extended objects, this is simply done by subtracting the 2-D data from one nod position from the other. For most datasets, you may be able use the **bksub.fits* spectra from the *pbcd/* directory in your data extracted from the archive. These files are difference of the stacked (**coa2d.fits*) products for each nod position). For the data in this moving target example, there are no **bksub.fits* products so we will do the background subtraction by hand. This may be useful in other cases, if you have further pre-processing you would like to explore.

```
;idl program to perform sky subtraction on IRS data
filename1='SPITZER_S2_6068992_0004_6_E2949446_coa2d.fits'
filename2='SPITZER_S2_6068992_0005_6_E2949448_coa2d.fits'
filename3='6068992_LLO2_4_5.fits'
filename4='6068992_LLO2_5_4.fits'
r1=readfits(filename1,hdr1)
r2=readfits(filename2,hdr2)
r3=r1-r2
r4=r2-r1
writefits,filename3,r3,hdr1
writefits,filename4,r4,hdr2
end
```

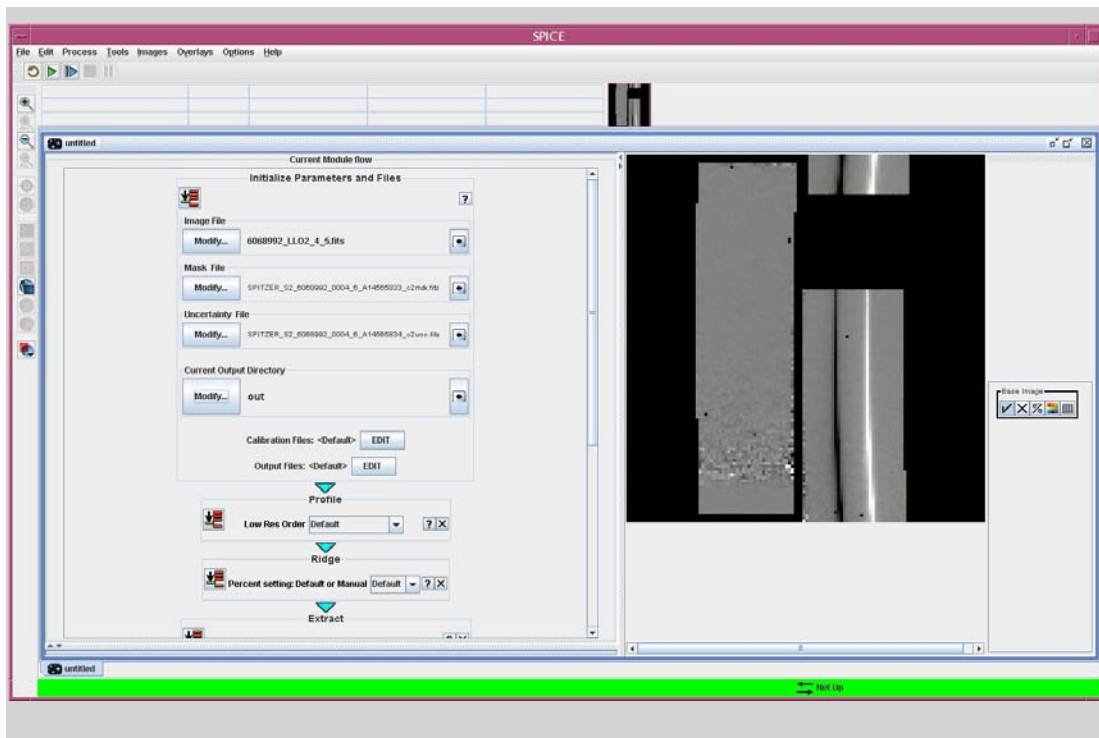
You can coadd the individual DCE's yourself (taking care that you coadd only those exposures at a given slit nod position). There is no Spitzer-specific tool to do the sky subtraction, and we suggest you use your favorite data-reduction software to do this, ensuring that the sky-subtracted output is written out as a FITS image (and retains the FITS header corresponding to one of the DCEs at the proper nod position). For example, in IDL:

2. Launch the SPICE GUI. For example from the UNIX command-line :

```
unix% spice &
```

Within the SPICE GUI, open a point source extraction flow using **File --> Open Spice Generic Template --> Point Source with Regular Extract**.

3. Load your data into SPICE by specifying the input data directories and names in the **Initialize Parameters and Files** module. Begin by clicking the "Modify" button for the "Image File". Provide the input background-subtracted 2-D spectral image (which you created in step 1), and the corresponding uncertainty image, and mask file. Normally, the corresponding mask and uncertainty files will be automatically selected, but we have made our own image file, so we must enter them manually. If you used the **coa2d.fits* files to create your sky-subtracted 2-D image, then use the *c2unc.fits* and *c2mask.fits* files from the *pbcd/* directory for the positive nod position, as your uncertainty and mask files for the sky subtracted data.



You can change the contrast ("stretch") of the 2-D image you are viewing using the color-grid button on the left of the FITS window, and using the display functions.

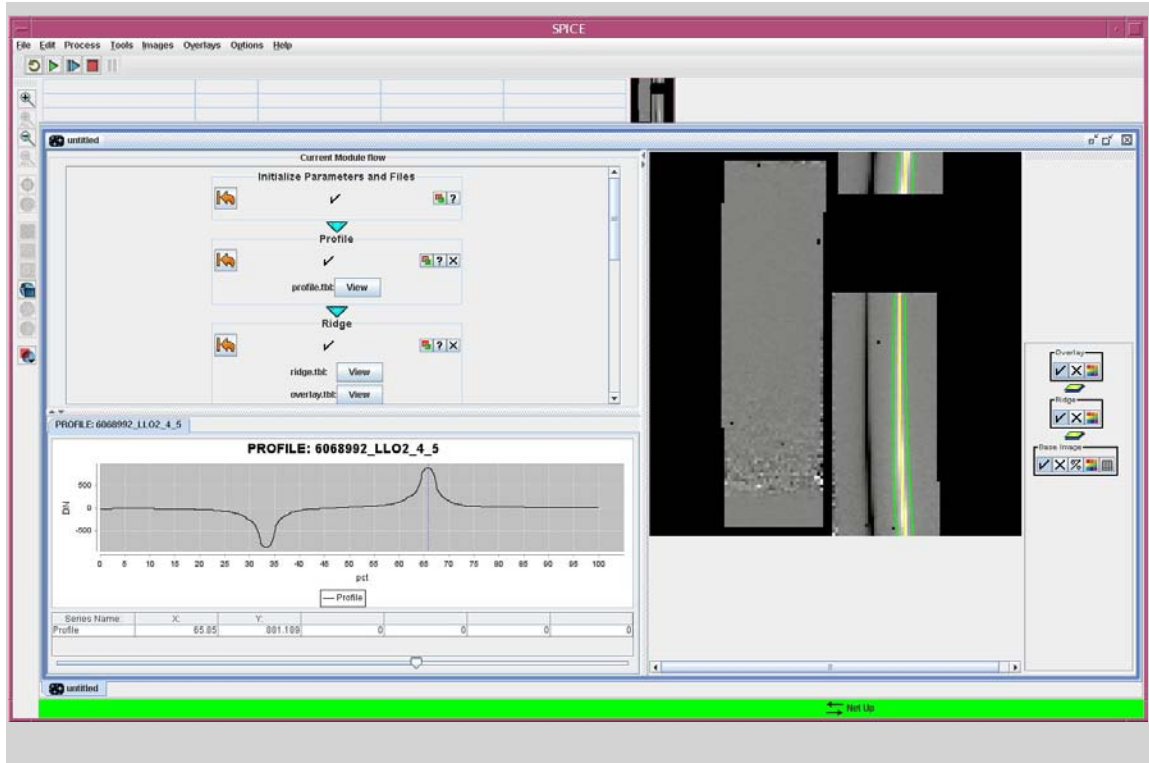
4. You can check the calibration files which SPICE will use with the "Calibration Files" button. Normally, you will allow SPICE to auto-select the pipeline version. Alternatively, as shown below, you can point SPICE to the custom calibration (CAL) files if you are an expert IRS user or if SPICE updates fall behind releases of new calibration by the SSC.

5. You can select the output directory and the names of the output files, using the corresponding "Modify" and "Edit" buttons in the input module. The output **.profile.tbl*, **.ridge.tbl*, **.extract.tbl*, and **.spect.tbl* files generated by the extraction will be written to the specified directory. The ultimate output table of interest is the **spect.tbl* -- this is your extracted 1-D spectrum in (ascii) IPAC table format. These files will be overwritten if the flow is rerun, so you may wish to change the output file names if wish to have multiple versions in a common output directory.

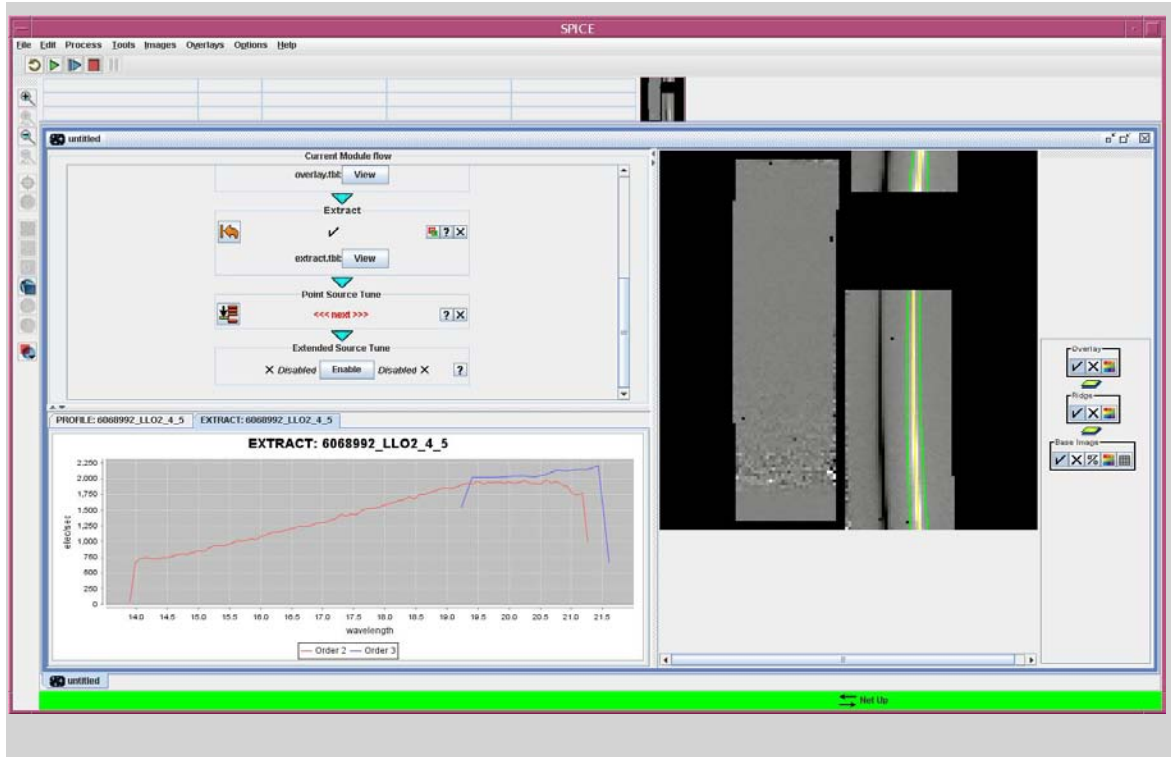
When you have set all the input parameters, run the **Initialize Parameters and Files** module, using the run button in the upper left of the module window.



6. To begin the extraction, Run the **Profile** module, which calculates the wavelength-collapsed average profile in the spatial direction across the 2-D background-subtracted spectrum. (Although it is not appropriate in this example, the **Profile** module can be set to the non-target spectral order instead of using the default, which would then be the one extracted). Note that for the background-subtracted spectrum, a "positive" profile and a "negative" profile (from the other nod) relative to the zero level, are seen as the output. Next, establish the (peak) ridgeline of the spectrum in the dispersion direction along the 2-D spectrum by running the **Ridge** module. You can either allow SPICE to automatically derive the ridge peak or set the ridgeline peak manually by using the slider on the profile plot or typing into the box provided (clicking on the plot will move the dotted percentage line but not change the value in the box). When the module is run, the ridge line and the extraction window will be overlaid on the FITS image.



7. To extract the spectrum, run the **Extract** module. The extraction can be done with "Default" settings. For the low-resolution modules, the spectrum is extracted along the Ridge location, in accordance with the wavelength-dependent Point Spread Function (PSF) and the spectral trace (with "Auto" width). Alternately, the **Extract** function can employ a window with a different width ("Manual" width), but the width will still scale with wavelength, unless a full-slit extraction ("Full" width) is specified (using the ExtSrc option under "Width"). Note here that an "extended source" is expected to fill the entire slit. Although it is possible to manually set the extraction width (e.g. change it to 7 pixels at 14 microns) to pick up the comet's coma, the default flux calibration in the subsequent step will not be accurate. Note that the output of the extraction is still in instrumental units, i.e., electrons/sec, and we perform the conversion to flux in the next step.



8. You still need to "tune" the extraction by applying the flux conversion from instrumental to absolute flux units. To do this, run the **Point Source Tune** module. Again, in this case the definition of an extended source is one that fills the entire slit so we use the Point Source function. The module corrects the slope and curvature of each order by applying the polynomial coefficients in the *fluxcon.tbl* calibration file. This correction is based on an order-by-order comparison of calibration data to standard star model spectra. The flux units are now in Janskys (Jy). Note that this conversion is only accurate for point sources and so will work for the comet nucleus, using the default ridge parameters. Currently, the tune calibration will not be accurate for extended targets that do not fill the slit, such as the nucleus + coma in this case. If you have MIPS or IRAC photometry of the same extended target you observed with IRS, it may also be possible to cross-calibrate the spectra using these imaging observations, as was done for the SW-1 observations (Stansberry et al., 2004, ApJS, 154, 463).

9. SPICE will output the profile, ridge and extract information, and the 1-D spectrum, as table files. This will go to the output directory you specified in step 5.

10. To quit SPICE, pull down the "File" menu and select "Exit."

Recipe 17. SPICE: Spectrum Extraction of ultra-deep IRS data

In this recipe, we demonstrate the data reduction strategies used for ultra-deep IRS spectroscopy for faint 24 micron sources. The ultra-deep proof of concept data (PID=252) included observations in SL1, LL2, and LL1. We use the LL1 data here. The target was a $z=2.69$ ULIRG. A total of 12 hours on-source integration was performed, using spectral mapping mode to place the target at 6 positions along the slits.

This recipe includes some preprocessing before using SPICE. We assume that you are comfortable with some flavor of standard image processing software such as IRAF or IDL. We will describe the basic procedures and trust that they are straightforward to implement in your favorite package.

17.1 Requirements

- Download and install SPICE (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/spice/>).
- For a better understanding of the wavelength range and names of the modules see the IRS pocketguide (http://irsa.ipac.caltech.edu/data/SPITZER/docs/files/spitzer/irs_pocketguide.pdf).
- Using the Spitzer archive tool, download the BCD data associated with the LL2 observations in AORs 15733248, 15732992, and 15732736. The files of interest are in the *ch2/bcd/* subdirectories, specifically the **bcd.fits*, **bmask.fits* files. See Recipe 1 for instructions on using the Spitzer archive.

17.2 Pre-Processing

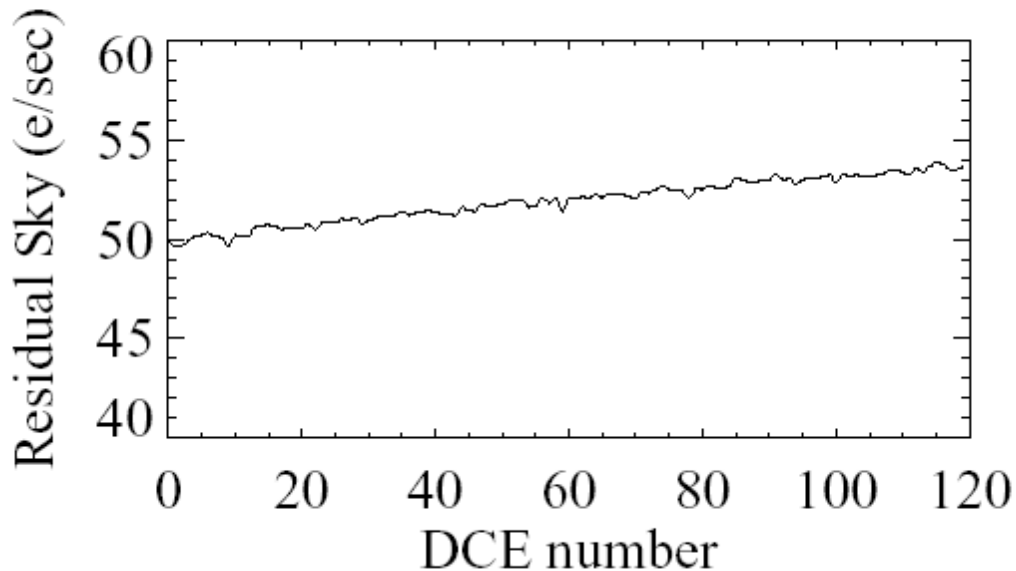
The basic calibrated data (BCD) files have been processed by the Spitzer pipeline, which includes ramp fitting, dark subtraction, droop correction, linearity correction, flat fielding, and wavelength calibration. Further processing of the two dimensional dispersed frames is required before spectral extraction:

17.2.1 Latent charge removal

A small fraction (1-2%) of charge on the detector persists between frames despite the resetting of the detector that occurs prior to every integration. This latent charge decays slowly over time and is removed by the annealing process. In the case of very faint sources, the source of latent charge is the zodiacal background. Over the course of a six hour AOR, this charge can build up to a significant level. You can remove the latent signal by fitting the slope of the background with time.

Measure the background independently for each row of each frame as the median of that row. Next, fit the slope of those median values and subtract that amount row-by-row from each frame. The background in

BCD images is the residual after "skydark" subtraction in 3D. In the deep observations, the residual sky level is about 50 electrons per second, and the latent charge built up to about 5 electrons per second by the end of the AOR (see figure below).



The latent charge accumulated in a pixel as a function of time (DCE number). The latent charge was measured in the median of 400 pixels.

17.2.2 Rogue Pixel Interpolation

Unstable, or "rogue", pixels are those which are useable in some AORs and not in others, depending on the recent history of the detector. A mask of known rogue pixels is provided by the SSC, but you can identify further suspect pixels from the data. In many LL observations, rogue pixels are not too much of a problem, because they tend to be subtracted away. The very long time baseline of the current dataset means that they can still show up even after subtracting a delta sky (see below).

We recommend using IRSCLEAN (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/irsclean/>) to find rogue pixels in the 2D data. Using the **Agressive** option, you can have the program flag more pixels than are in the SSC mask. You can further identify suspect pixels by searching for pixels with abnormally high variance with time (inspecting them in IRAF or IDL, not within IRSCLEAN). Finally, though it may be unsatisfying, some visual inspection is often used to find any rogues that still sneak through. Once you have a list of rogue pixels, they can be interpolated over using IRSCLEAN.

17.2.3 Residual Sky Subtraction

Once the BCD data are "cleaned", you will need to create residual sky images from the data. The observations were taken in mapping mode, so a good sky frame can be created using the other five map positions. A median combination is reasonable, although a resistant mean (removing outliers before calculating the mean) with time is also a good way to calculate the sky value in each pixel. The sky should be calculated for each AOR separately, to keep the frames close in time. If the sky appears too noisy, it is also possible to combine skies from multiple AORs taken close in time. After sky frames are created, they should be subtracted from each BCD frame. In some cases, you may wish to scale the sky frame to the mode of the sky in the LL1 order for best results.

17.2.4 Combine the stack of each Map Position

Once the frames have been corrected for latents, cleaned of rogue pixels, and sky subtracted, they can be combined into final frames. Each map position should be combined separately, because the spectra are not rectilinear on the array thus preventing shift-and-add. Again, a resistant mean is a reasonable approach to combining the frames. The final frame will need a valid FITS header, so attach it to, for example, the first header in the stack (unless your software propagates the header for you).

17.2.5 Measure the Uncertainty Frame

Given the high redundancy for each pixel in this ultradeep observation, you can create a new uncertainty frame, rather than propagating the BCD uncertainties. You can estimate the uncertainty in the final, combined frames as the standard deviation of each pixel with time. Save this uncertainty frame along with the final frames to be input into SPICE. Again, make sure you have a valid FITS header.

17.2.6 Combine the Bmasks

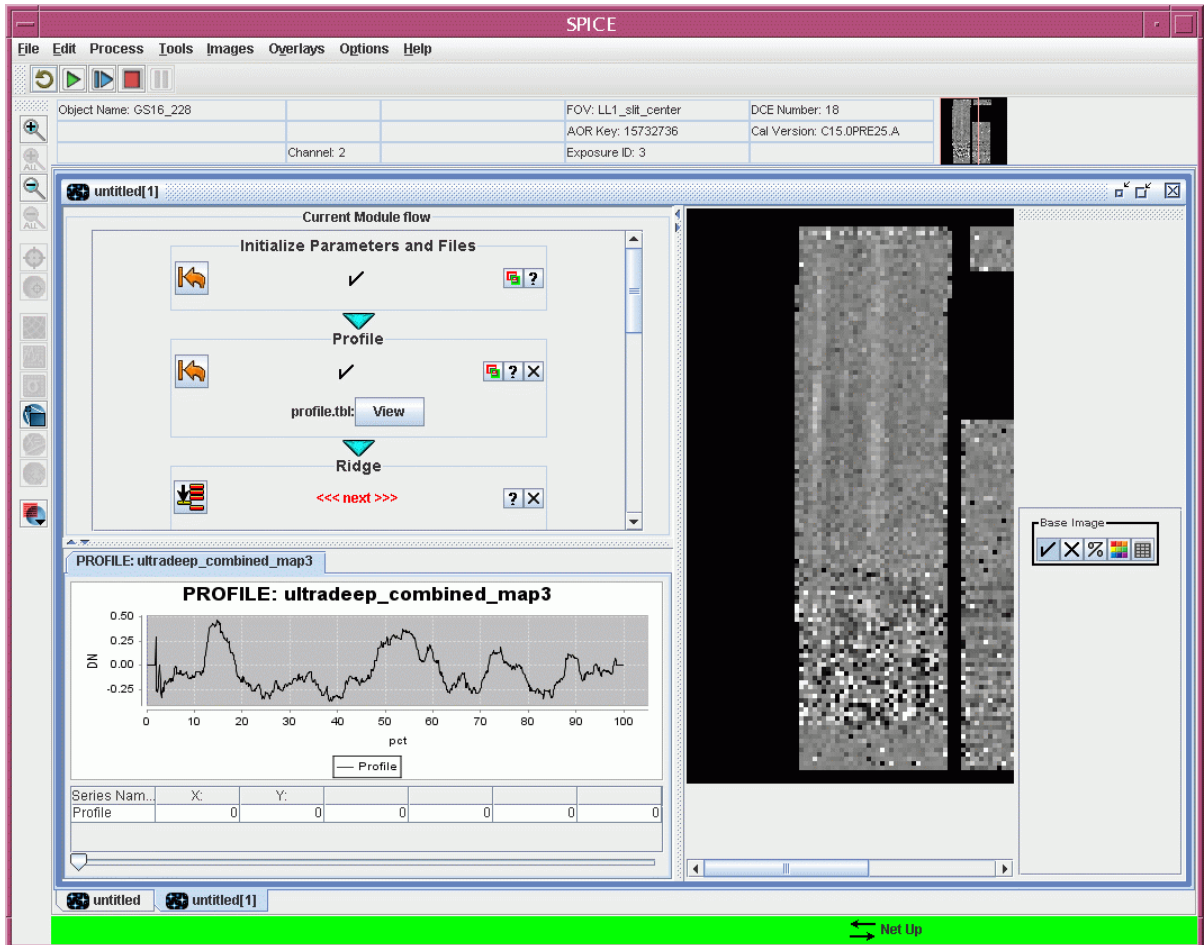
In principle, the *bmask.fits* files should be combined along with the BCD frames. This should be done with a bit-wise AND. The high redundancy and rogue pixel cleaning means that few transient pixel problems will survive and so only permanently bad pixels will need to be excluded from the final extraction. For a quick look, you could even use just one of the mask files and see what happens.

17.3 Optimal Extraction with SPICE

You will need to extract the spectrum from each of the map positions independently. We will walk through the extraction of the third map position as an example here. Repeat the same analysis for the other positions. Given the faint continuum of the object, it is usually not possible to do this in batch mode (the RIDGE module won't find the trace). Launch the SPICE GUI, and choose **Open Spice Generic Template-->Point Source with Optimal Extract** from the file menu. This will bring up the optimal extraction flow. Then:

1. Load the combined data, uncertainty and mask files that you created in the preprocessing step. You can specify an output directory at this step. Run just the **Initialize Parameters and Files** module using the stack button in the corner of the module. The dispersed image will be displayed in the FITS window.

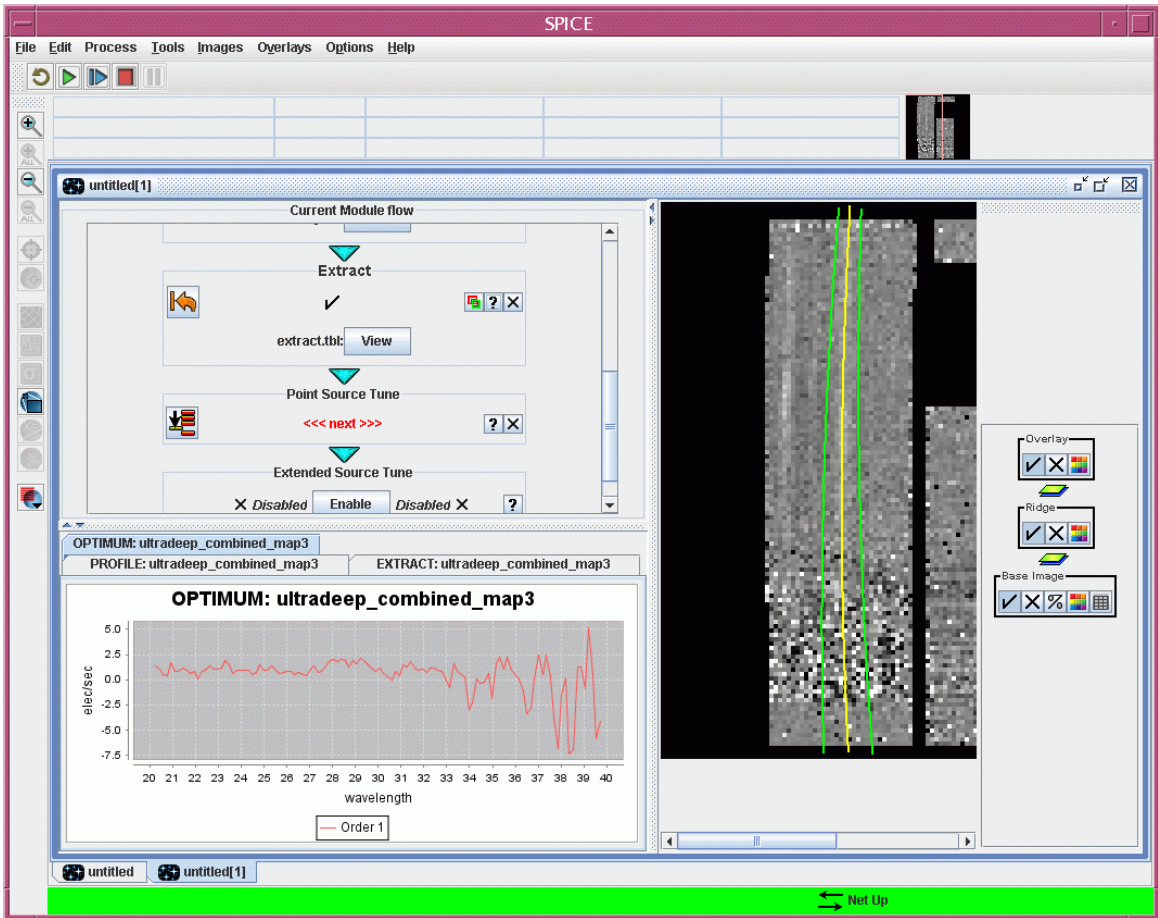
2. Run the PROFILE module. There is no need to change the order, so leave the setting as "default". After you click the stack button, the spatial profile will be calculated and displayed in the plot window (see figure below). The profile includes other objects in the slit, as well as significant scatter due to noise.



The SPICE GUI after running PROFILE for the third map position. Several objects can be seen in the slit. The object of interest is the middle one.

3. Because the target (in the middle of the slit in this case) is not the brightest object in the slit, you cannot run the RIDGE in the default mode. Instead, you will need to select "Manual". This will bring up the input field to specify the PERCENT across the slit at which you would like to place the RIDGE line. You can use the slider at the bottom of the plot window to move the percentage across, as indicated by a dotted blue line in the plot window. (Note -- Clicking the plot window will move the dotted line but not the percentage in the RIDGE module). When you have the dotted line in a reasonable spot, run the RIDGE module. The trace and the extraction window will be overlaid on the FITS image. It may take a few tries to find exactly the right percentage. To rewind the RIDGE module, just click on the rewind button that has replaced the stack button in the module window.

4. Now you can run the optimal extraction. You should not need to change the default parameters, as the optimal weighting will ensure that only the most significant pixels contribute to the extraction. The spectrum will be displayed in the plot window.



The SPICE GUI after running Optimal extract. The source is quite faint, but a very weak PAH feature can be seen at about 29 microns (redshifted 7.7 and 8.6).

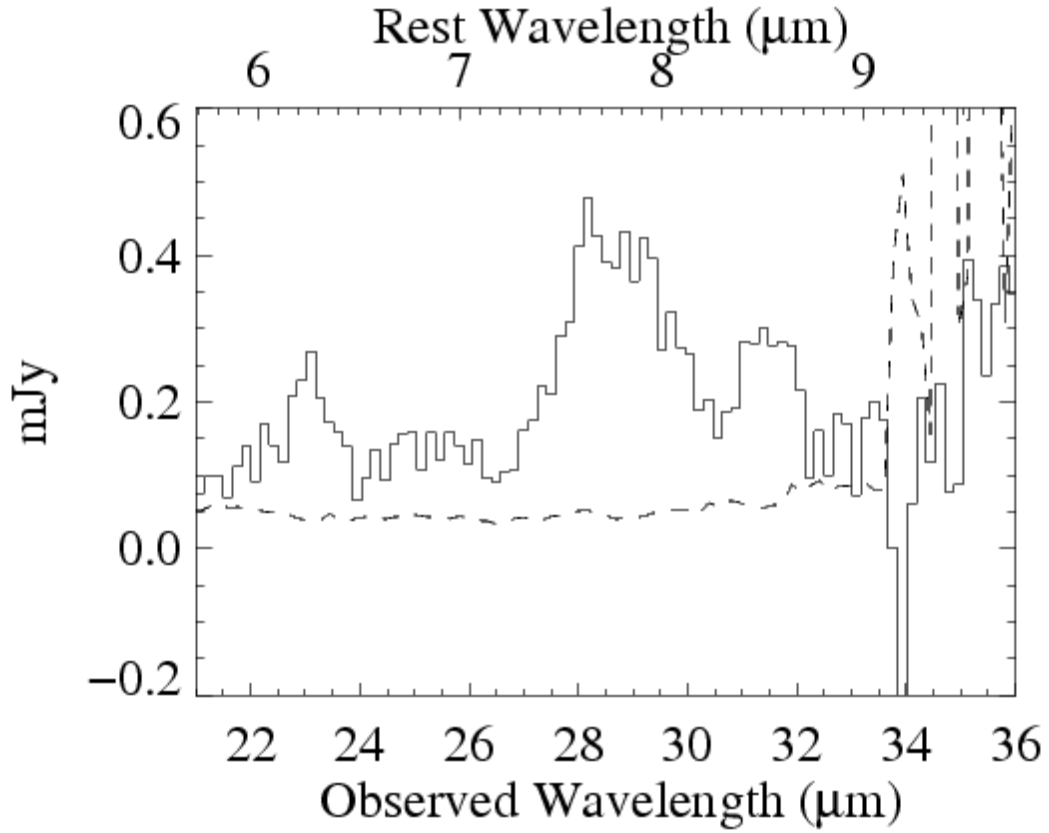
5. Finally, run the POINT SOURCE TUNE module to convert from electrons per second to Jy.

17.4 Post-Processing

After extracting the 1D spectrum from each of the six map positions, you will need to coadd them to obtain the final spectrum. The output of SPICE is a spectral table (in your specified output directory) **.spect.tbl*, where the *** is the name of your input FITS file with the *.fits* suffix stripped off. The output file is an IPAC table, with the following columns:

order	wavelength	flux	error	bit-flag
int	real	real	real	int
	micron			
		Jy	Jy	

The 1D spectra themselves can simply be averaged. The uncertainties should be added in quadrature.



The final spectrum, after averaging the six SPICE extractions. The 1-sigma error array is plotted as a dashed line. The target lies at $z=2.69$, so the wavelength scale is shown in both the observed and rest frames.

Recipe 18. IRSCLEAN, COAD, and SPICE: Mrk 273 LH Data

In this recipe, we describe how to reduce high-resolution IRS spectra using IRSCLEAN to remove rogue pixels, COAD to combine images taken at the same nod position, and SPICE to perform the extraction.

18.1 Requirements

To follow along with this recipe, you will need to use the following software:

- the [Spitzer Heritage Archive](http://irsa.ipac.caltech.edu/applications/Spitzer/SHA/), which is a web-based tool that does NOT require any installation on your computer. You can find it at <http://irsa.ipac.caltech.edu/applications/Spitzer/SHA/>.

- [IRSCLEAN](http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyistools/tools/irsclean/), which is an IDL tool that you can download at <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyistools/tools/irsclean/>. To run IRSCLEAN, you must have IDL version 5.5 or later installed.

- [SPICE](http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyistools/tools/spice/), which is a Java-based tool that can be downloaded at <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalyistools/tools/spice/>.

18.2 Download and Examine the Example Data Set

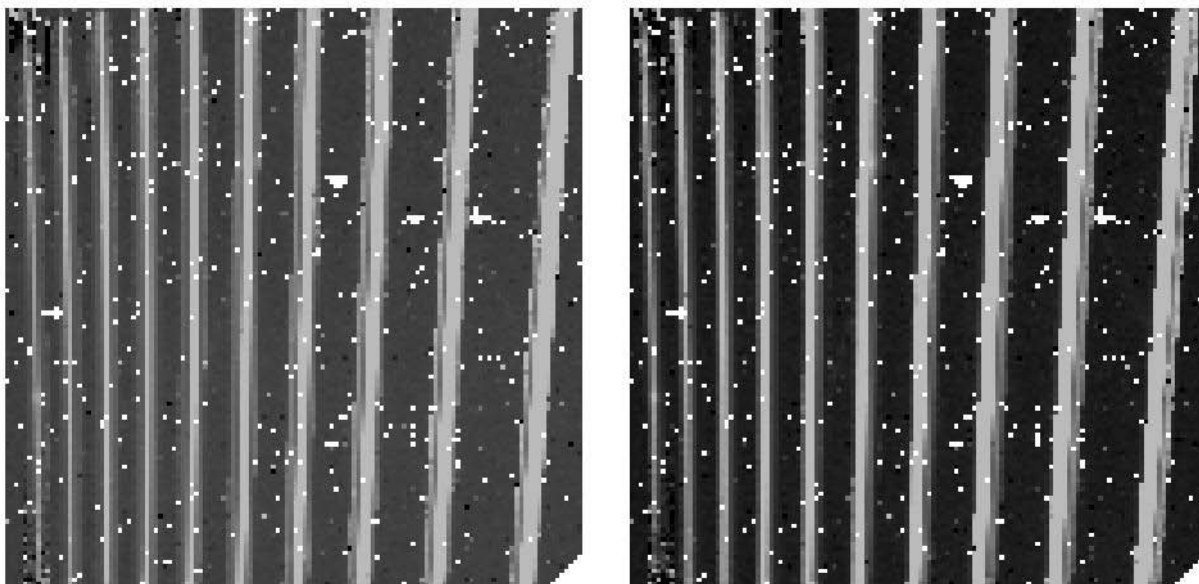
The example data set that we will use for this recipe consists of high-resolution IRS Staring-Mode observations of the nearby ULIRG Mrk 273 at a redshift of $z = 0.038$. Consult Recipe 1 and Recipe 2 to learn how to use the Spitzer Heritage Archive to download AOR 4980224. For this recipe, you will need the BCD (Level 1) and PBCD (Level 2) data for the Long-High module (channel 3).

Once you download the data to the directory of your choice (which we will now refer to as *yourdirectory*), you should examine the files in the directory

```
yourdirectory/r4980224/ch3/bcd/
```

At the time of writing of this recipe, the data for this AOR were reduced with Spitzer pipeline version S18.7.0. Check the header keyword 'CREATOR' in any of the files in the above directory to see which pipeline version your downloaded data set was reduced with. Depending on the results, you might expect to see small differences from the results presented here and those you achieve with your data set.

By examining the header keyword "FOVNAME" in the **bcd.fits* files in the above directory, you will notice that there are four nod 1 files (FOVNAME = 'IRS_Long-Hi_1st_Position') and four nod 2 files (FOVNAME = 'IRS_Long-Hi_2nd_Position'). In the image below, we show an example of each.

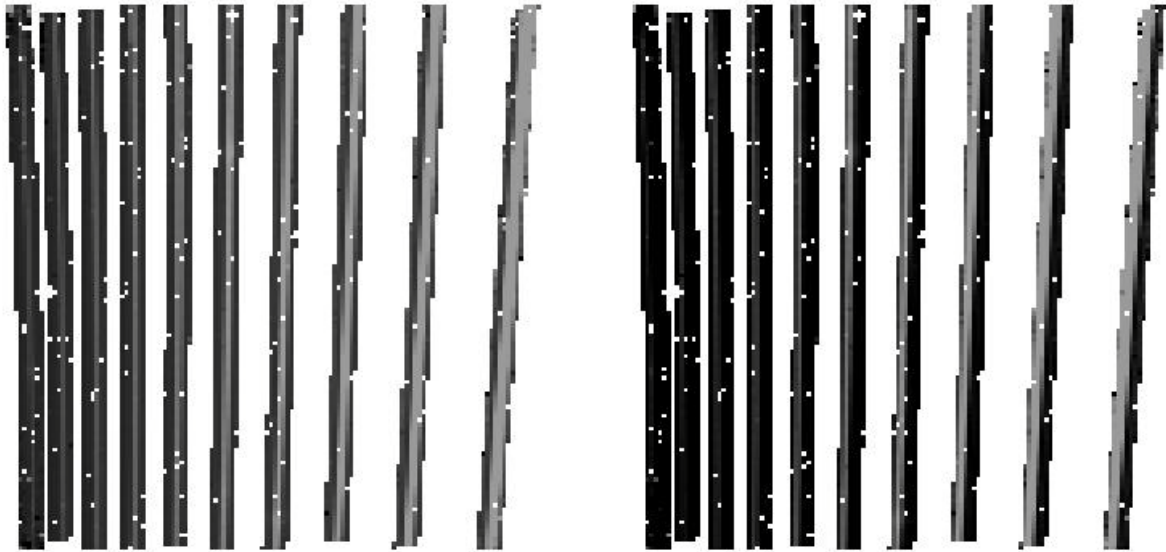


Example nod 1 (left) and nod 2 (right) BCD images of Long-High (channel 3) IRS spectroscopy of the ULIRG Mrk 273. There are four BCDs taken in each nod position for this AOR. You can see the light from Mrk 273 illuminating the different orders, with the spectral trace shifting leftwards from nod 1 to nod 2. Also visible are many hot pixels. An important part of our reduction is to ensure that those hot pixels do not impact our extracted spectra.

It is also useful to examine the files in the Post-BCD directory, which is:

yourdirectory/r4980224/ch3/pbcd/

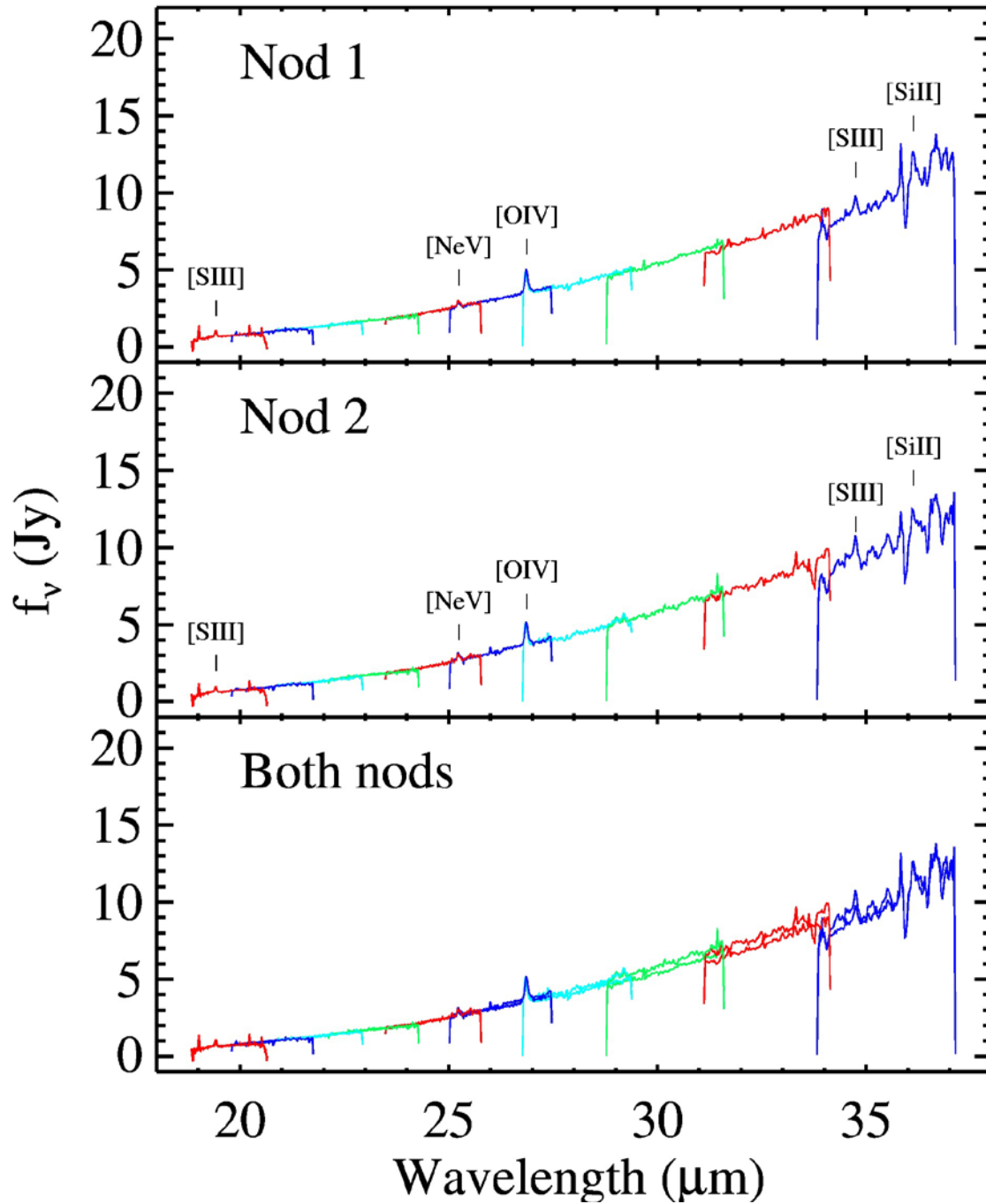
The relevant files here are the two **coa2d.fits* files, which are the co-added nod 1 files and the co-added nod 2 files, with some additional masking between the orders. These files are illustrated below.



Nod 1 (left) and nod 2 (right) **coa2d.fits* files for Long-High (channel 3) IRS observations of Mrk 273. Each was constructed from four BCD images. Hot pixels remain along the spectral trace. This indicates that we may need to do some hot pixel cleaning before spectral extraction.

Before proceeding with the reduction, it is also worth examining the **tbl* files in the *pbcd/* directory. These are ascii tables that contain the extracted spectra. Using the headers within the files for guidance as to their contents, you can use your favorite graphing program to plot the spectra, as we have done below.

Mrk 273



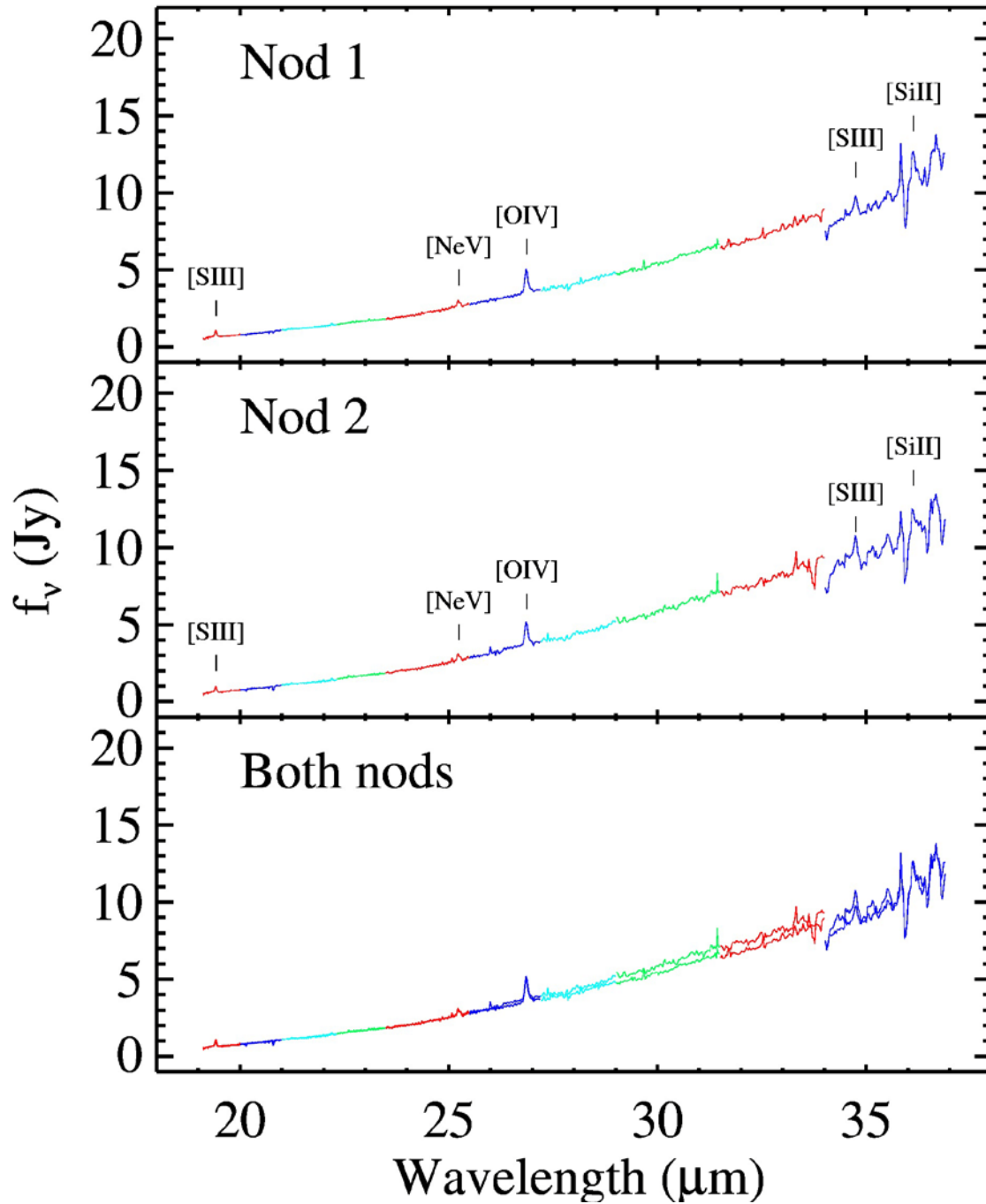
Long-High spectra of Mrk 273, as taken from the pbcd/ directory. This represents the simplest reduction possible, as it was accomplished by an automated pipeline. Adjacent orders are plotted in different colors for clarity. No cleaning or trimming of the spectra has been performed.

The first thing we notice about the spectra above is that the edges of the orders tend to plummet. This is easily solved by trimming the orders, since there is a small but significant overlap between orders. Users are free to use their judgment when trimming, but we use the values below:

order:	11	33.8268	37.1360 -->	34.0	36.9
order:	12	31.1221	34.1383 -->	31.5	34.0
order:	13	28.7773	31.5925 -->	29.0	31.5
order:	14	26.7647	29.3831 -->	27.2	29.0
order:	15	25.0177	27.4652 -->	25.5	27.2
order:	16	23.4824	25.7797 -->	23.5	25.5
order:	17	22.1149	24.2784 -->	22.3	23.5
order:	18	20.8966	22.9408 -->	21.0	22.3
order:	19	19.7948	21.7494 -->	20.0	21.0
order:	20	18.8294	20.6542 -->	19.1	20.0

This simple trimming makes for a much smoother spectrum. However, it's obvious that the order mismatch is large for the lowest orders (longest wavelengths).

Mrk 273



Same as the previous image, but with some trimming of the orders.

Some real spectral lines have been indicated in the above plots. Noise, hot pixels, and detector fringing (which may be improved using [IRSFRINGE](#)) remain. In the next section, we will investigate and remove the contribution from hot pixels.

18.3 Remove the Rogue Pixels Using IRSCLEAN

Assuming that you have IRSCLEAN installed, you can simply type the following at the IDL prompt and have it guide you through the process of cleaning your BCD images.

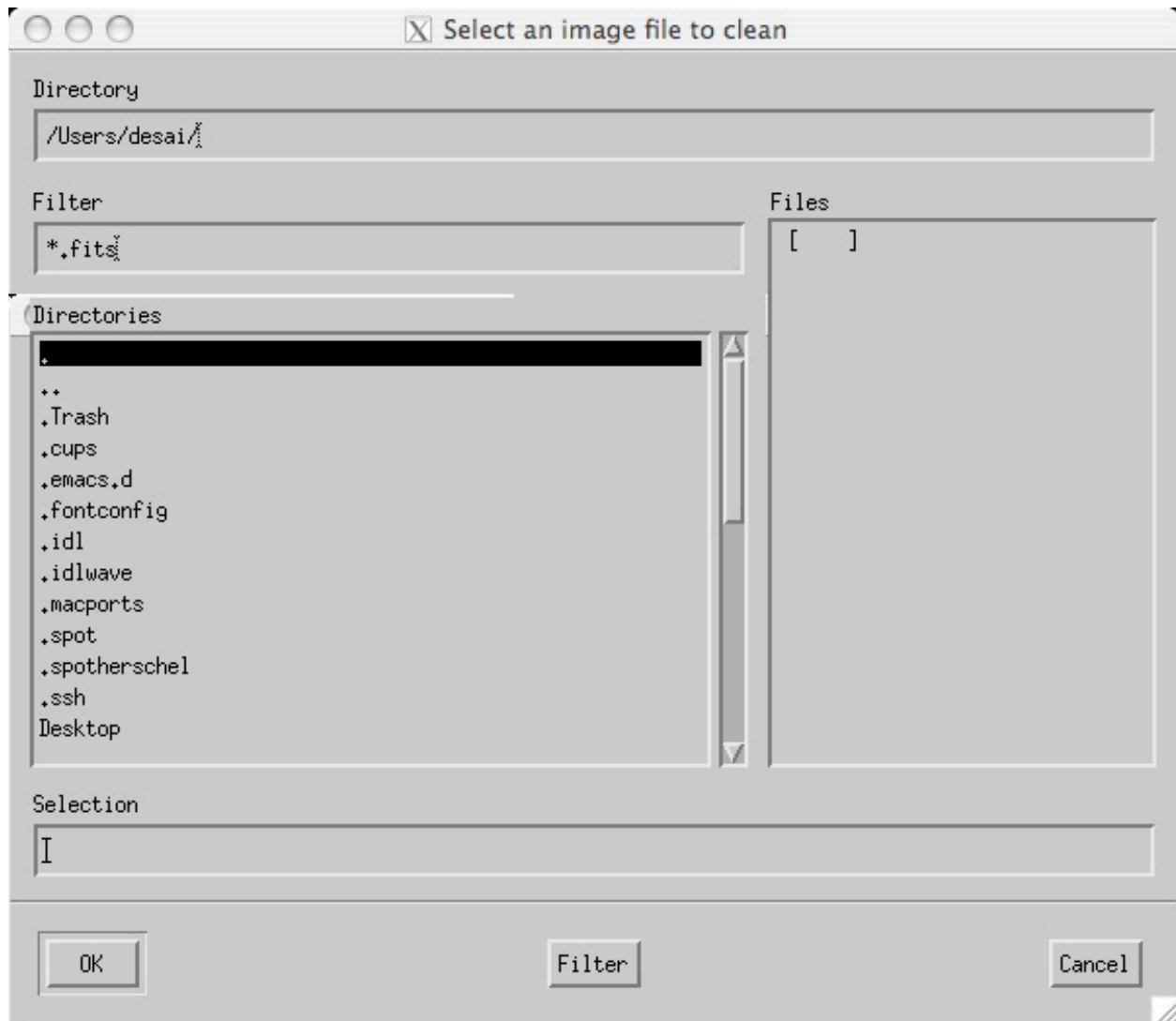
```
IDL>irsclean_mask, datarange=[0,0]
```

The `datarange` parameter describes the minimum and maximum pixel values used in the display of the fits image. By setting it to `[0,0]`, we will be able to interactively adjust the display when we examine and (optionally) edit the mask. The `datarange` parameter is one of many command-line options allowed by IRSCLEAN. For a full listing, type the following at the IDL prompt:

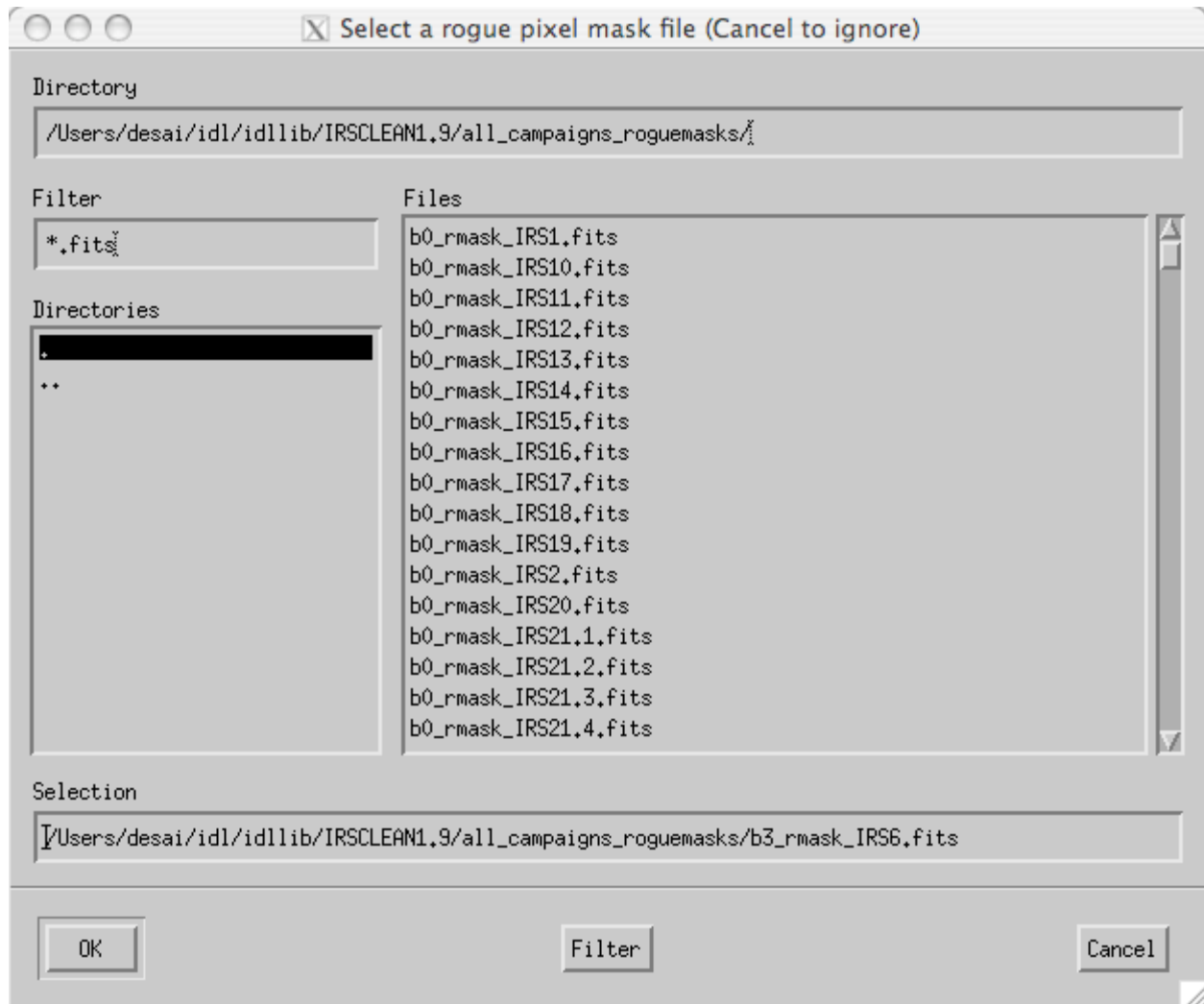
```
IDL>irsclean_mask, /help
```

Using a minimum set of input parameters makes IRSCLEAN interactively walk you through the cleaning process.

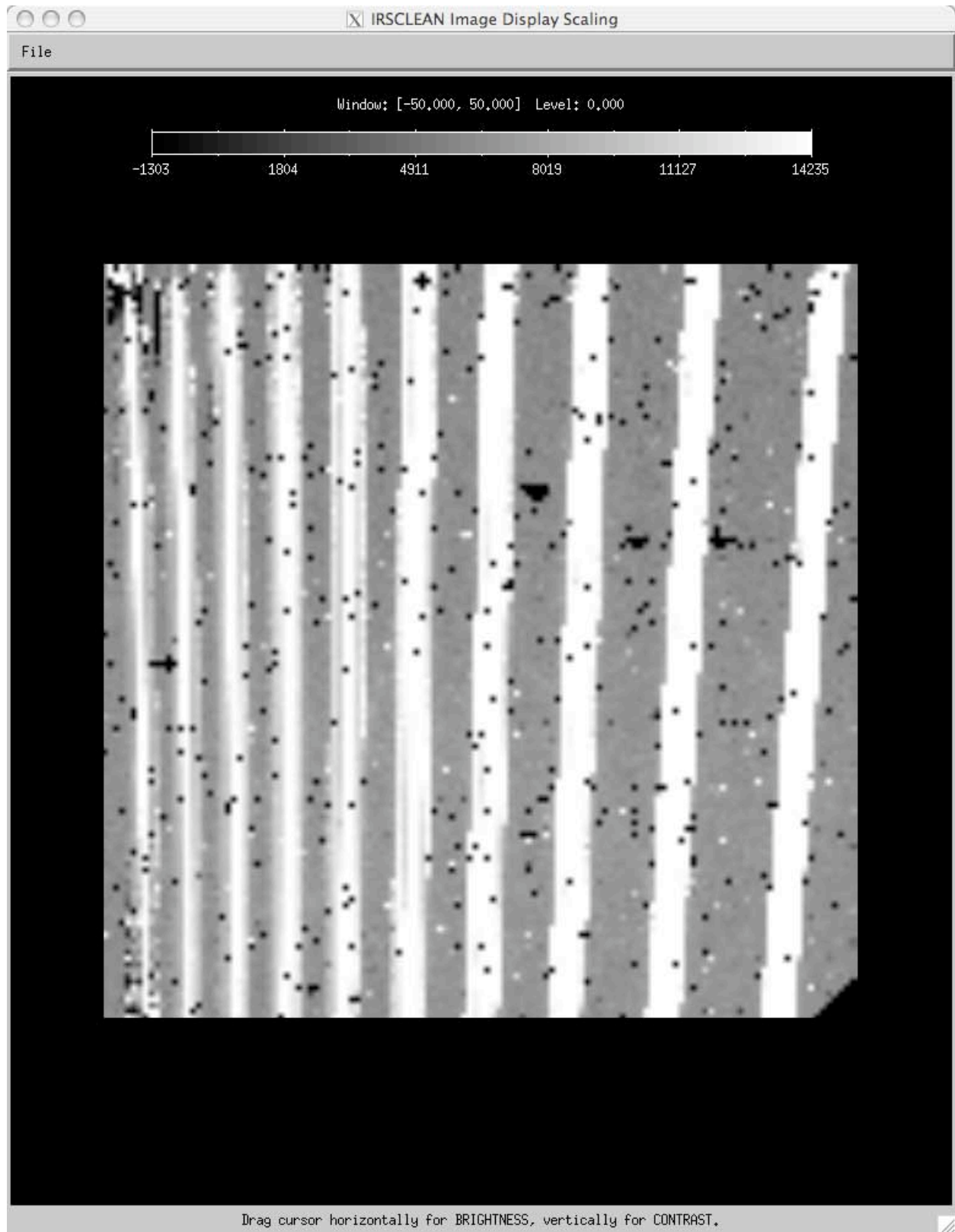
Once you issue the first command above, you will see the following screen pop up:



You can change the filter to read **bcd.fits* to get a list of just the **bcd.fits* files. Select one of these and click "OK". You will then be asked to select a rogue pixel mask file:



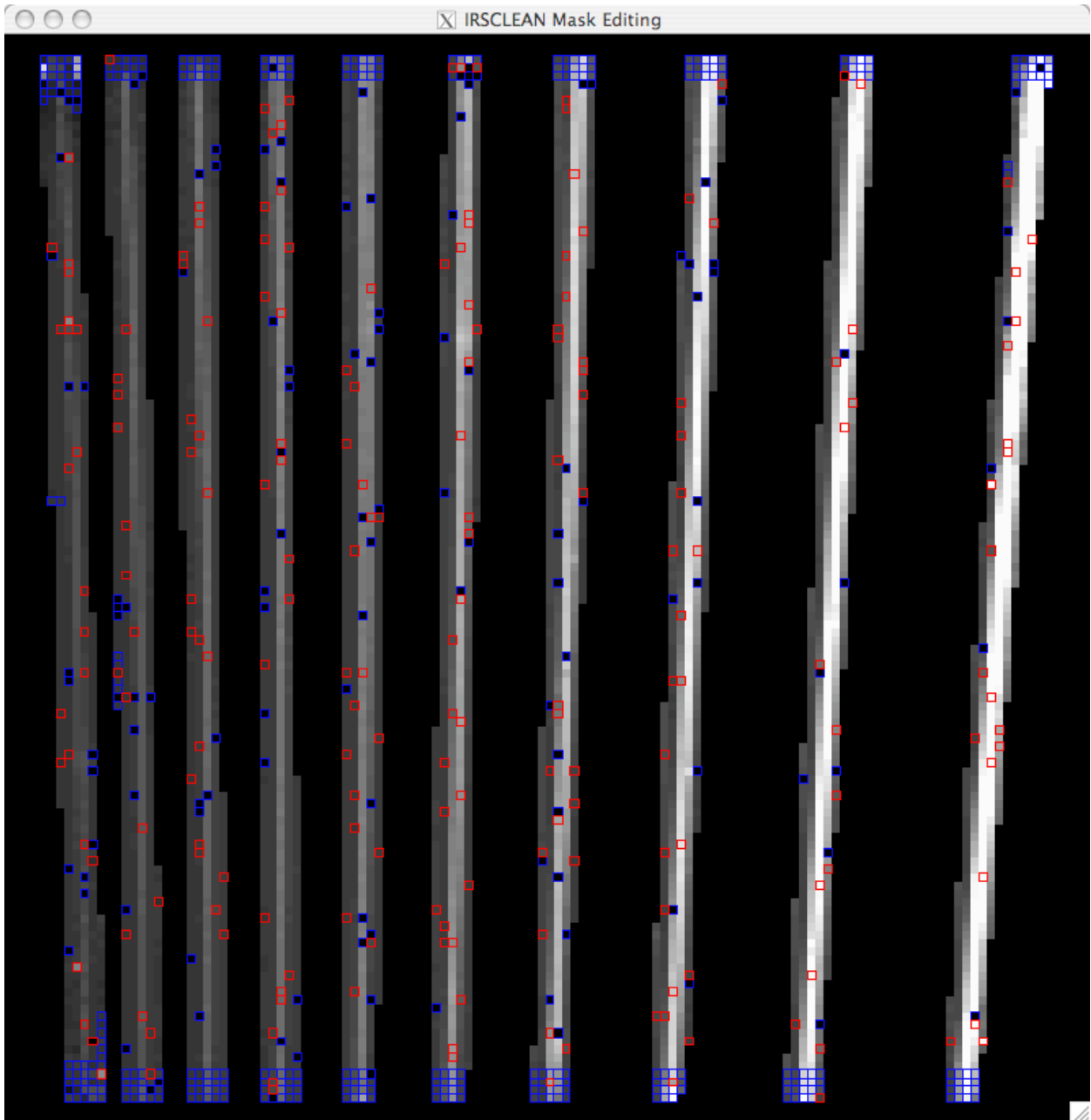
IRSCLEAN will choose a default mask, which is what you want. Click "OK". (Alternatively, you may select "Cancel" to select no rogue mask.) Note that the keyword /CampaignRmask automatically picks the campaign rogue mask, bypassing this dialog widget. Next you will be presented with the IRSCLEAN Image Display Scaling screen:



As printed at the bottom of this screen, you can right-click on this screen and drag the cursor around until you achieve the kind of display you desire. The vertical white stripes are the different orders of the spectrum. The greyscale is set so that the white parts are illuminated. The black pixels are NaNs, and will not be included in the extraction by SPICE. You want to adjust the display until it becomes obvious which pixels are anomalous (other than the NaNs). You only need to worry about the white illuminated strips. The interorder regions won't be used in the extraction. It can be difficult to find a datarange that fits all of the orders, but you can rescale the image during Mask Editing (see below) by typing the "r" key. This allows you to go through the process multiple times, doing a few orders at a time.

Once you have chosen a display setting that you like, go to "File"--"Quit" to proceed with the mask editing process.

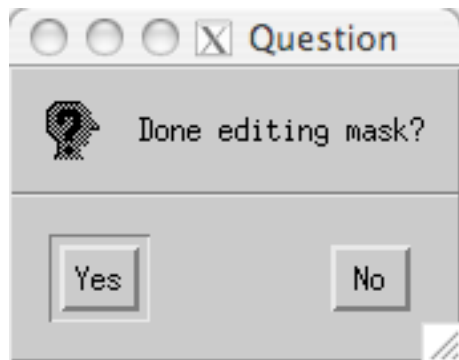
You will be presented with the IRSCLEAN Mask Editing screen, which looks like this:



In the terminal in which you started IRSCLEAN, you'll see these instructions:

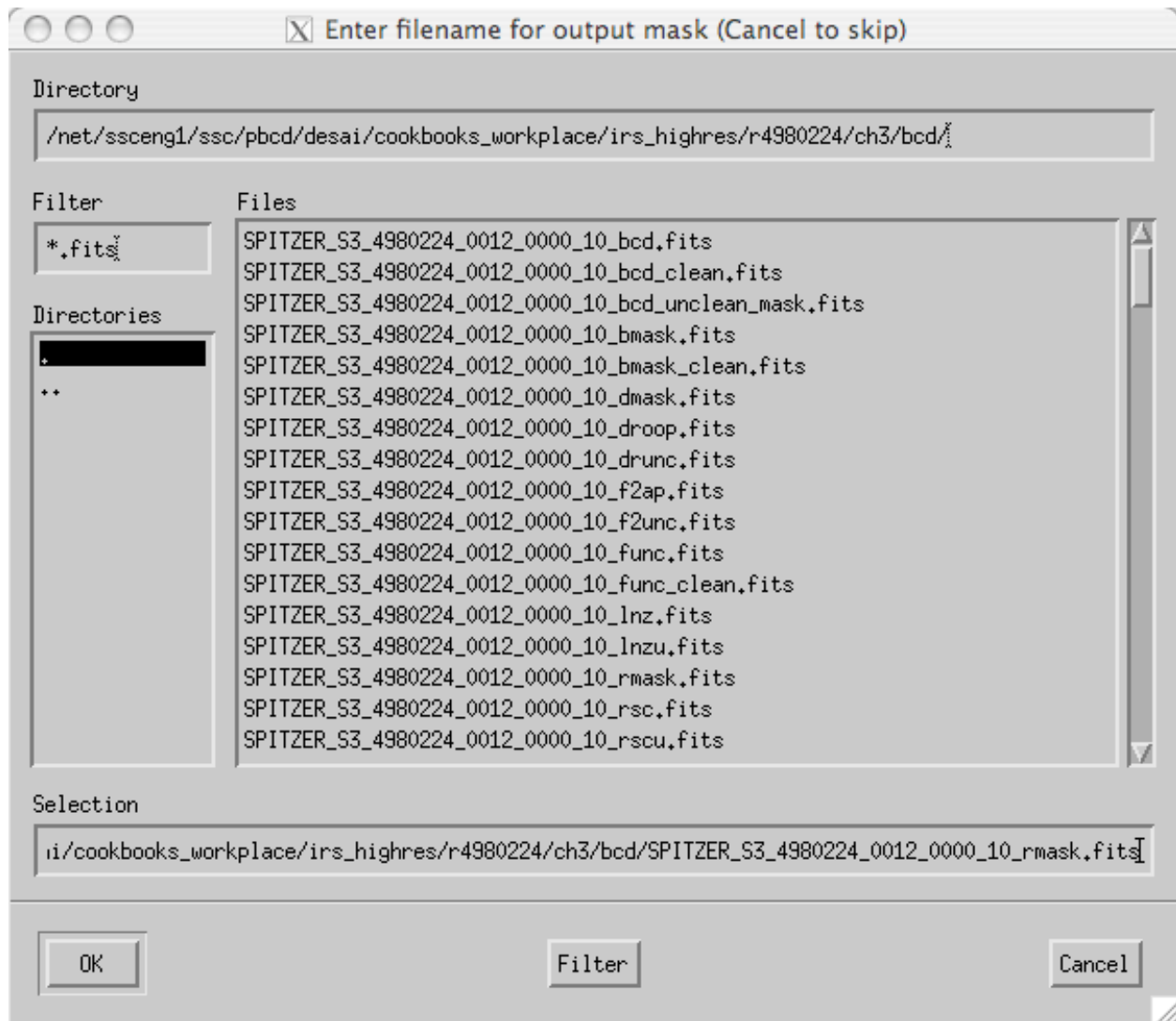
```
*****  
  
STAGE 1: MANUAL MASK-EDITING  
  (requires 3-button mouse)  
  
Blue = bmask-ed pixel (ignored in cleaning and rogue finding)  
Red = Rogue pixel  
  
LEFT  - toggle rogue mask pixels on and off.  
MIDDLE - quit without saving.  
RIGHT - clean data using current mask  
  
*****
```

The bmask is the mask that is associated with the **bcd.fits* image, and includes pixels that were flagged throughout the pipeline run. You will notice that all of the NaNs (black pixels) are shown in blue, which means they are flagged in the bmask. The campaign rogues from the mask that you uploaded into IRSCLEAN are shown in red. These include anomolous-looking pixels (too bright or faint compared to neighboring pixels). At this point you can follow the directions above to edit the mask. For the purposes of this example, we will simply use the campaign rogues as is. To do this, you can just right click on this window (or type "c").



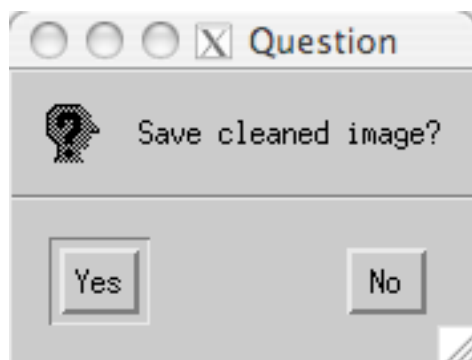
When the screen above shows up, click on "Yes".

Next you will be prompted to enter the filename for the output mask:



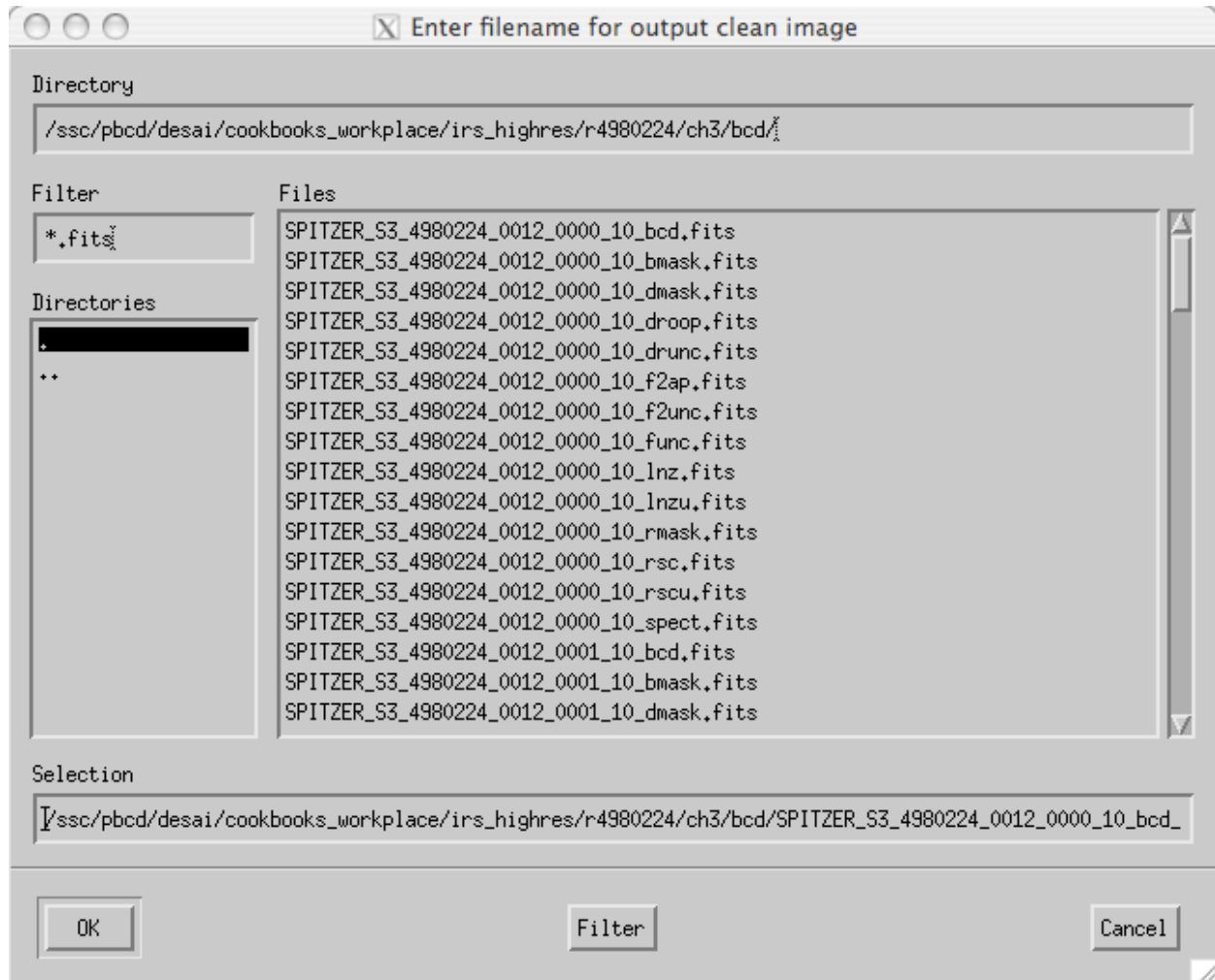
IRSCLEAN will choose a default value that will work fine. Click "OK".

Now you will be asked if you wish to save the cleaned image:

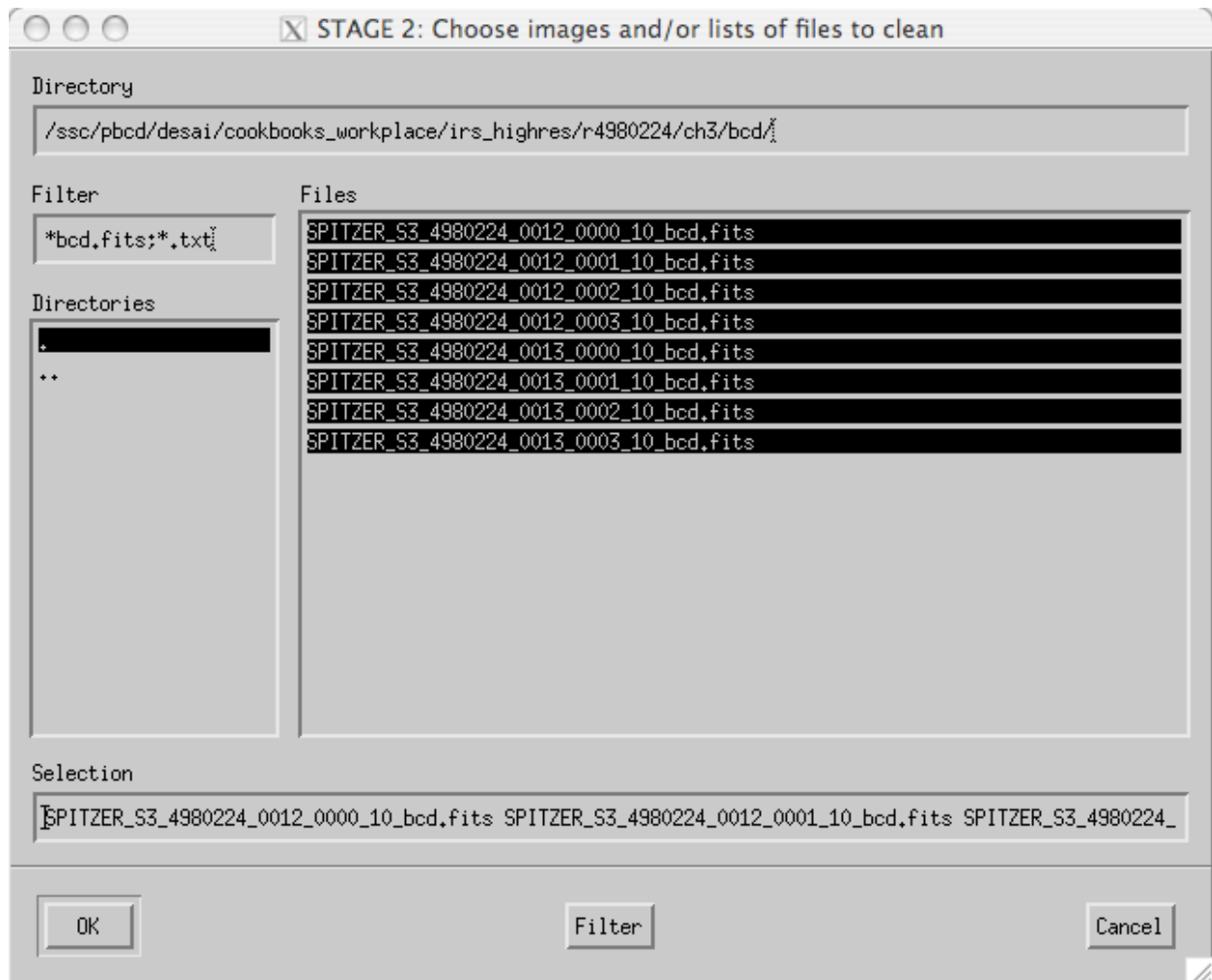


Click "Yes".

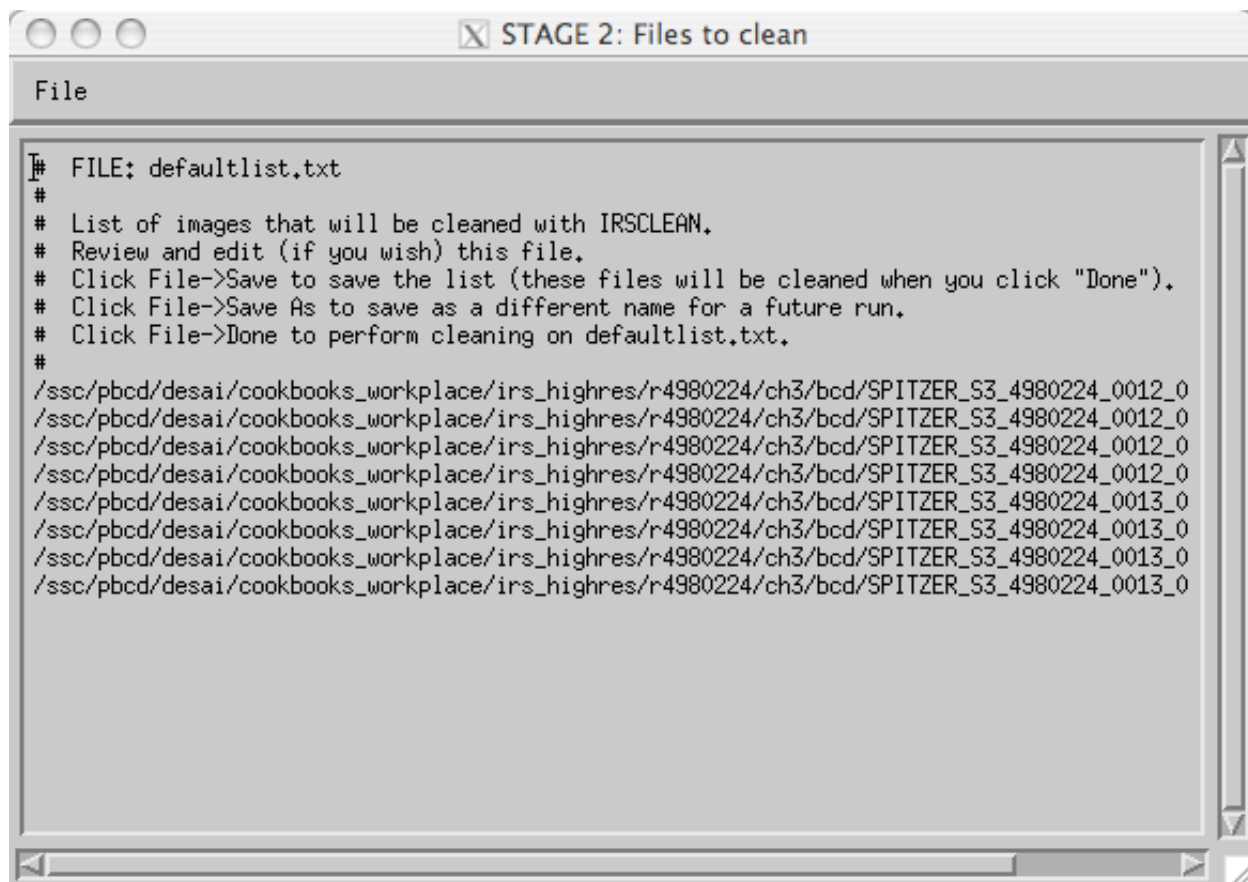
Now you will be asked to enter the filename for the output clean image:



IRSCLEAN will choose a default name that will work fine. Click "OK". Now IRSCLEAN will offer to clean other images based on the same mask. Since you didn't edit the campaign rogue mask, you can use it to clean your other **bcd.fits* files. Simply edit the Filter field to read **bcd.fits*. Then choose all of the bcd files by holding down the control key for each selection. Finally, click "OK".



Now you will be presented with a summary of images that you just chose:



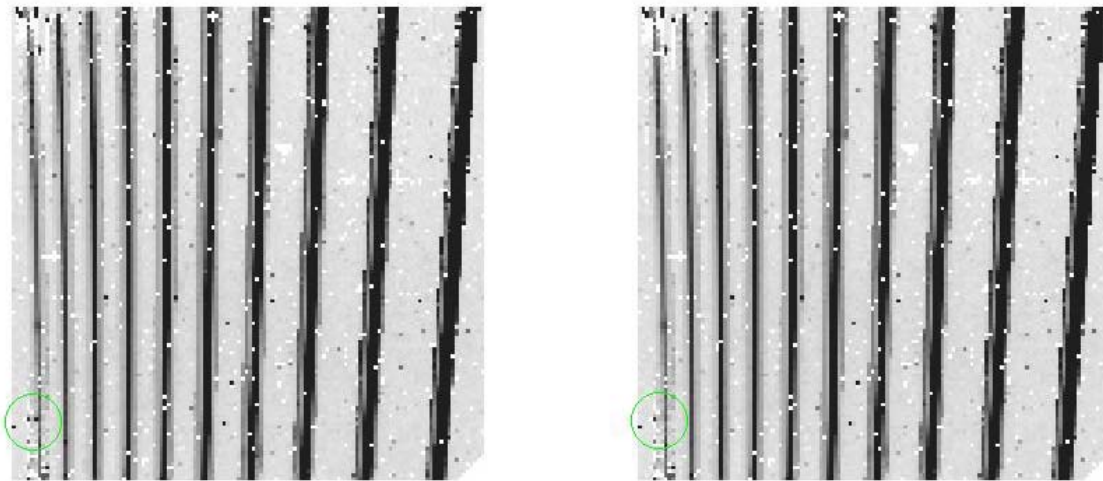
If what you see is correct, go to the "File" menu and choose "Done with Stage 2".

You will see a lot of output to the terminal window in which you started IRSCLEAN. At the end, it will tell you that you could have performed the same task by typing the following command at the IDL prompt:

```
IDL> irsclean_mask,
'r4980224/ch3/bcd/SPITZER_S3_4980224_0012_0000_10_bcd.fits',
inRmask_File=['IRSCLEAN1.9/all_campaigns_roguemasks/b3_rmask_IRS6.fits
'],
outRmask_File='r4980224/ch3/bcd/SPITZER_S3_4980224_0012_0000_10_rmask.
fits',
outClean_File='r4980224/ch3/bcd/SPITZER_S3_4980224_0012_0000_10_bcd_cl
ean.fits', filesToClean=['r4980224/ch3/bcd/defaultlist.txt'],
DataRange =[-181.434,979.094]
```

You have now just cleaned your **bcd.fits* images based on the campaign rogue masks. You will find the results in your *bcd/* directory. The most important of these are the **bcd_clean.fits*, **bmask_clean.fits*, and **func_clean.fits*.

At this point it is a good idea to compare the original **bcd.fits* images to their cleaned versions (**bcd_clean.fits*), and to make sure you are happy with the results.



Original **bcd.fits* image (left) and the corresponding **bcd_clean.fits* image (right). The images look mostly the same, but differ in detail. The green circles show one example where the images look different. One of the rogue pixels near the spectral trace has been removed. Other changes exist in the images, and the user is encouraged to inspect the fits images closely to discover them.

18.4 Coadding your BCD files

The real test of whether the cleaning process has been worthwhile is to compare the final spectrum with the uncleaned spectrum that we previously examined from the *pbcd* directory. Before we can do any extraction, we must first coad the **bcd_clean.fits* files taken at each nod position. You can do this using the IDL software COAD, which can be found at

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/coad/>.

To use COAD, first create a list of the cleaned bcds that you wish to coad.

```
% ls *bcd_clean.fits > mixedbcdlist.txt
```

This list will contain both nod 1 and nod 2 positions, but COAD will figure it out and produce one coadded file for each position. To run COAD on this mixed list, type the following at the IDL prompt:

```
IDL> doall_coads, 'mixedbcdlist.txt', /verbose, fatalbits=[8,12,13,14]
```

The result cleaned, coadded files and their associated uncertainty and mask files are:


```
SPITZER_S3_4980224_0012_10_coa2d.fits
SPITZER_S3_4980224_0013_10_coa2d.fits
```

```
SPITZER_S3_4980224_0012_10_c2unc.fits
SPITZER_S3_4980224_0013_10_c2unc.fits
```

```
SPITZER_S3_4980224_0012_10_c2msk.fits
SPITZER_S3_4980224_0013_10_c2msk.fits
```

At this point it is useful to blink back and forth (using ds9) between the the coadded images you have just created and the corresponding images in the *pbcd/* directory (e.g.

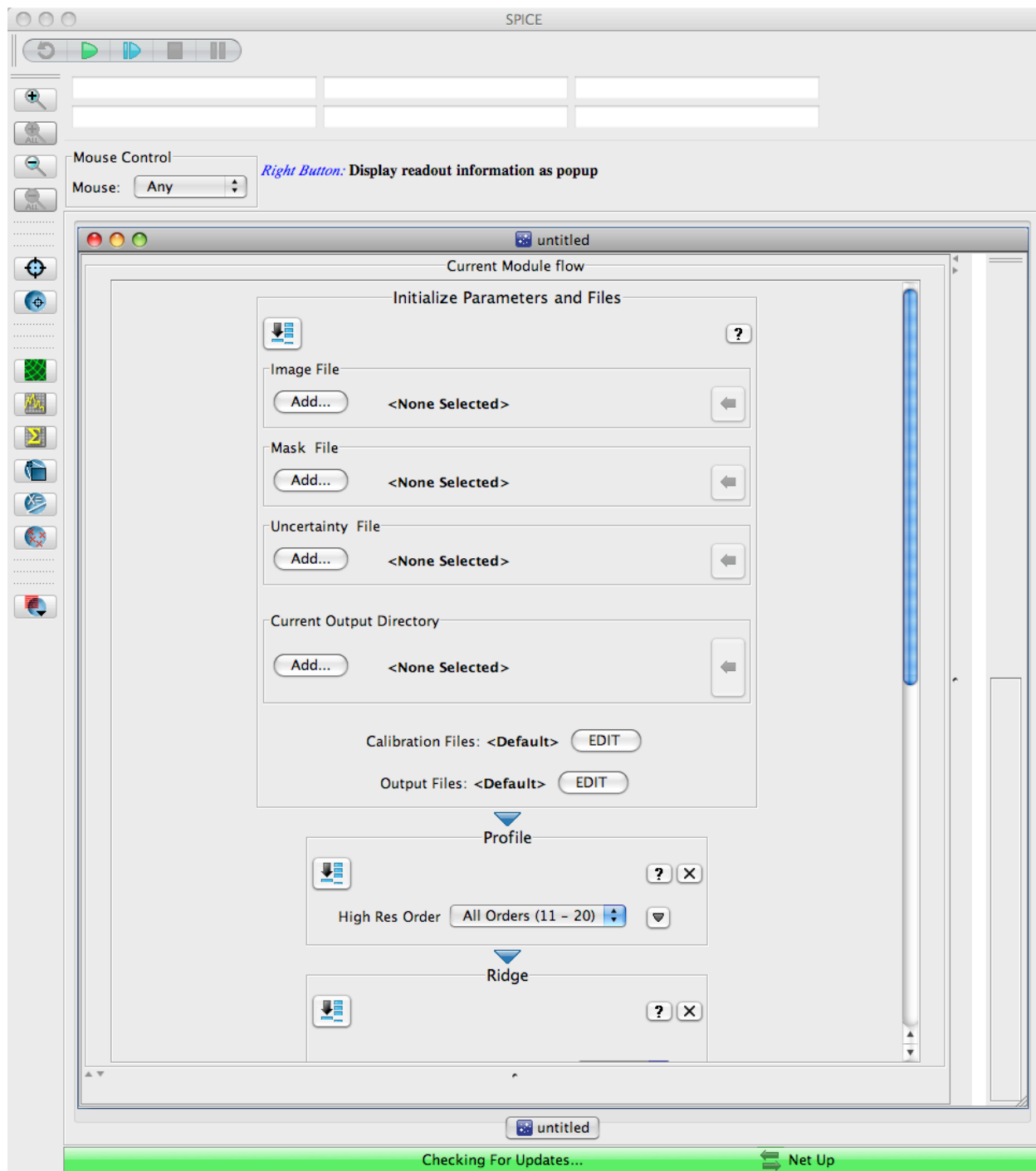
SPITZER_S3_4980224_0012_10_coa2d.fits versus

SPITZER_S3_4980224_0012_10_E5480030_coa2d.fits). The most noticeable difference will be that the *pbcd* pipeline masks the interorder regions, and COAD does not. If you take a more detailed look, you should see some anomolous pixels in the *pbcd/* image that you have removed in your image.

18.5 Extract the Spectrum Using SPICE

Now we can use SPICE to extract spectra from our coadded images.

Start up SPICE and then go to "File"-->"Open SPICE Generic Template"-->"Point Source with Regular Extract". You will see the following screen:



You must fill in all of the values requested near the top, under "Initialize Parameters and Files". For "Image File", choose *SPITZER_S3_4980224_0012_10_coa2d.fits*. SPICE should automatically find the corresponding mask and uncertainty files. Then choose the output directory of your choice.

Next click on the green arrow at the top of the SPICE window. When the flow has completed, you will be able to see your extracted spectrum at the bottom of the SPICE window. You will find the ascii version of this spectrum in the output directory that you chose, with a suffix of **spect.tbl*.

To extract the other nod, rewind the flow to the beginning, put in the nod 2 **coa2d.fits* file, and run the flow again. Alternatively, you can open a 2nd flow for the other nod. That way you can click back and forth between the 2 nod spectra.

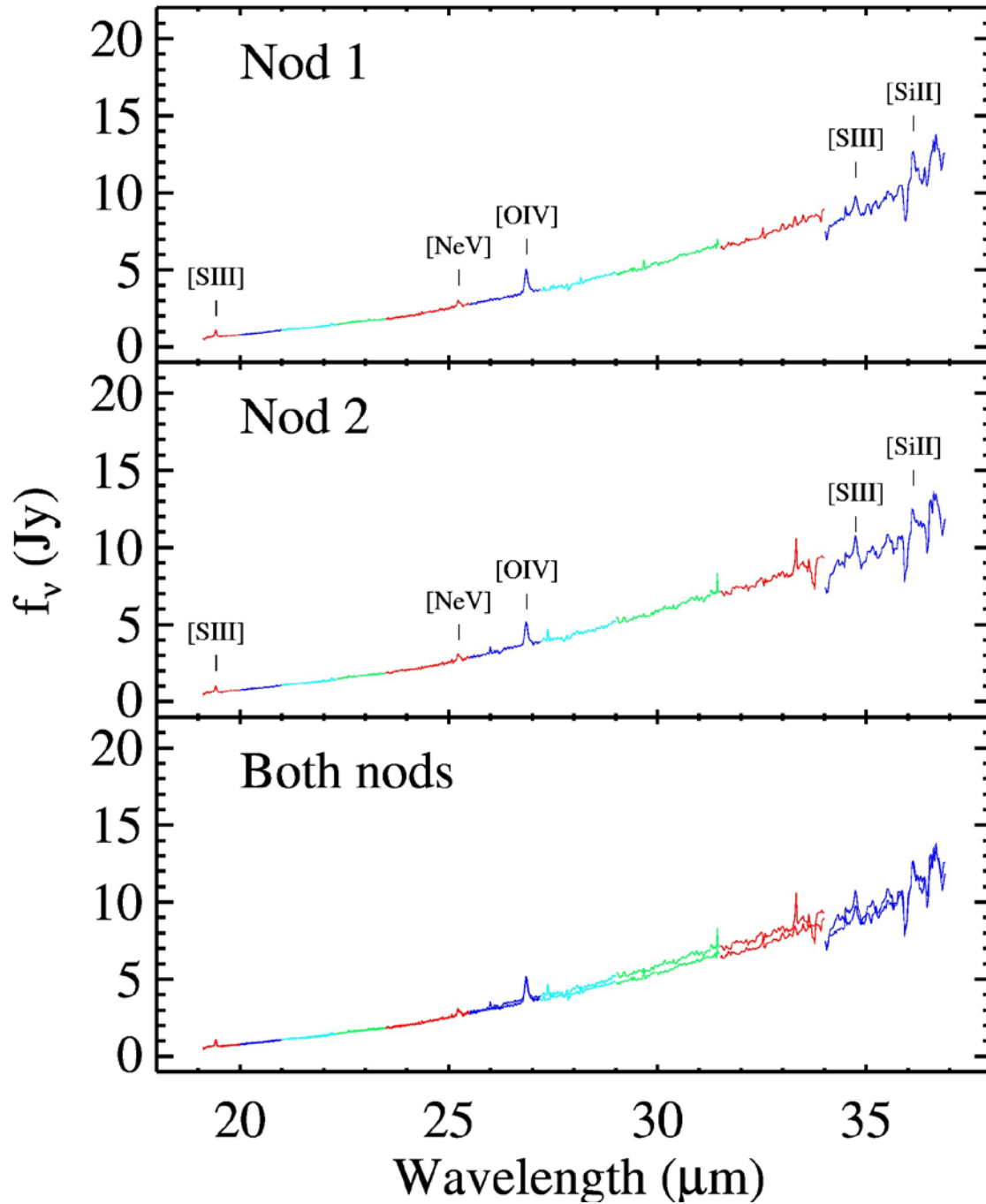
Now you have your own cleaned versions of the nod 1 and nod 2 spectral extractions:

SPITZER_S3_4980224_0012_10_coa2d.spect.tbl

SPITZER_S3_4980224_0013_10_coa2d.spect.tbl

You can plot these up in the same way you did before:

Mrk 273



You will notice that the spectra look very, very similar. However, if you perform a detailed comparison, you will find small differences. The largest differences occur at the longest wavelengths. In this case, there are two lines in the longest-wavelength order, so it is worth cleaning the spectrum prior to extraction. You may wish to go back and pay particular attention to this order and edit your mask accordingly.

So that you can check your results against ours, we provide our cleaned, extracted spectrum below:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/SPITZER_S3_4980224_0012_10_coa2d.spect.tbl

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/SPITZER_S3_4980224_0013_10_coa2d.spect.tbl

Recipe 19. SMART: General Steps for Extracting Staring Mode Spectra

19.1 About this recipe

The present cookbook explains how to extract IRS data for observations done in staring mode.

The cookbook is intended to be used with SMART v8.1.2 or later.

Check the online [online documentation](#) for any help on the SMART functionalities described in this cookbook. For an screenshot example of a basic extraction using optimal please go to the next recipe, which provides a screen shot example of an extraction.

19.2 Data preparation

- Create a new project and a new dataset. Remember to save the project after each step!
- Import the data into the dataset by browsing for `*_{bcd,func,bmask}.fits` files in the system.
- (optional): Select all the imported files and click on *Image operations -> 3-plane data -> Make 3-plane data*. This will new create `*_bcd3p` records which contain the uncertainty and BMASK information for each BCD image.
- **Cleaning:** Select all the sets of `{bcd,func,bmask}` records (or all the `bcd3p` records if the data was combined into 3-planes), and click on *Image operations -> Clean using IRSCLEAN -> Clean!*. Leave the default values and click on *Proceed*. If the cleaning is not satisfactory, remove the cleaned files and re-perform the operation with a lower mask value but the user must be aware that a too-low maskvalue can result in line clipping.
- **Co-adding:** Select all the cleaned records (`cl_*{bcd,func,bmask}` or `cl_*bcd3p`) and click on *Image operations -> Automatically combine DCEs for a given ExpID?*. This will combine the individual exposures.

19.3 Low-level rogue pixels removal

The data is now cleaned and the individual exposures are combined. However, bad pixels likely remain that were not cleaned. These are low-level rogues pixels that can be removed by subtracting an empty field image.

For this step, it is advised to use 3-plane data (see previous section). Three methods can be used:

- **Subtraction of a dedicated background observation (advised for high-resolution data):** select all the records (including the background) and click on *Image operations -> Image combinations & arithmetics -> Multiple subtraction*. Then select the records that should be removed from the others.
- **Subtraction by nod (only for low-resolution data):** for each module and order (SL1, SL2, LL1, LL2), select the 2 nods, and click on *Image operations -> Image combinations & arithmetics -> Arithmetics*. Then type the appropriate subtraction equation.
- **Subtraction by order (only for low-resolution data):** for each module and nod, select the 2 orders, and click on *Image operations -> Image combinations & arithmetics -> Arithmetics*. Then type the appropriate subtraction equation.

19.4 Spectral extraction

During extraction, the calibration files will be chosen automatically by SMART.

19.4.1 Point-like sources

High-resolution data: select the records and click on *Extract -> Full aperture (point-like source)*.

Low-resolution data:

- Before attempting the extraction, it is advised to run the [AdOpt](#) algorithm on at least one exposure of each module in order to check the actual extent of the source, the possible presence of multiple sources, and/or the presence of a significant extended background emission. The [AdOpt](#) GUI works with sets of 3 records rather than 3-plane data, so if the data is 3-plane at this point, select them, click on *Image operations -> 3-plane data -> Split 3-plane data*, select the new records and click on *Extract -> Optimal extraction (manual)*. Check the [AdOpt](#) help for more details on the use of the GUI.
- Two methods are available for extracting the low-resolution data:
 - Optimal extraction. An extraction is performed that uses the profile of the instrumental point spread function to weigh the data. It usually provides a better signal-to-noise ratio, especially for faint sources. Optimal extraction can be performed with the automatic source finder or with the manual source finder. The automatic optimal extraction will perform with no diagnostics available. A warning message will appear only if the source was found significantly off the intended position. To perform the automatic optimal extraction, select the records and click on *Extract -> Optimal extraction (automatic)*. The manual optimal extraction will open the [AdOpt](#) GUI. To perform the automatic optimal extraction, select the records and click on *Extract -> Optimal extraction (manual)*.
 - Tapered column extraction. A column extraction is used that scales with wavelength to account for the varying FWHM of the point spread function.

19.4.2 *Extended sources*

High-resolution data and low-resolution data: select the records and click on *Extract -> Full aperture (extended source)*.

19.5 Spectra visualization and analysis

Once the extraction is complete, the spectra will be displayed in IDEA automatically (except if manual optimal extraction was chosen, in which case the spectra are displayed in the [AdOpt](#) GUI. They can be exported to IDEA from there).

- **Displaying the spectra:** only the first spectrum is shown by default. To display another spectrum, click on the corresponding checkbox in the *Stored datasets* window, click on *Choose an apply a function*, and click on *Make prime dataset*. To overplot all the spectra, simply click on *Merge/overplot all datasets*
- **Order edges:** In IDEA, remove the order edges by clicking on *Order edges -> Zap order edges*. Important note: the new spectra is not yet stored, it can be by clicking on *Store Prime*.
- **Defringing:** LL1 spectra can require defringing. This can be done by clicking on *Defringe Tool*.
- Numerous functions are available in IDEA, such as line flux measurements, photometry measurements, blackbody fitting, spectra combination, etc... A wrapper for PAHFIT is also present. Refer to the [IDEA](#) help for more details.
- **Copy spectra to project manager:** the spectra can be copied to the project manager by clicking on *Copy to spectral manager*.
- **Saving spectra:** If desired the spectra can be saved in several formats (e.g. fits table, ascii table) in a given output directory. This can be done within the IDEA gui but it is easier performed once the spectra is copied into the project manager, and then clicking on *Export/Conver*. This step is not needed and the spectra remains in the project manager but the user must not forget to save the project before closing the session.

Recipe 20. SMART screen shot recipe

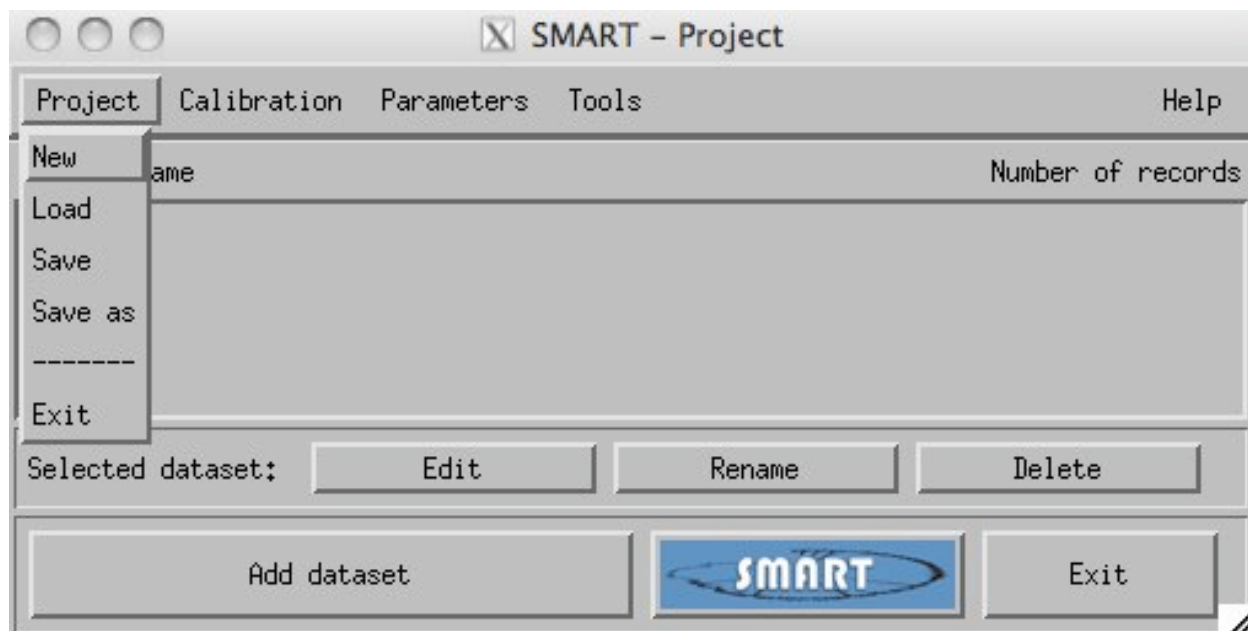
This example includes a step by step extraction of the source SMPLMC36 (AORkey=4949348), a planetary nebulae in the LMC. For simplicity this example includes only the extraction of the SL module. It includes snapshots of the GUI at every step. It does not explain the purposes of the different steps as this is explain in the previous recipe. Remember that we recommend to save the project at each step.

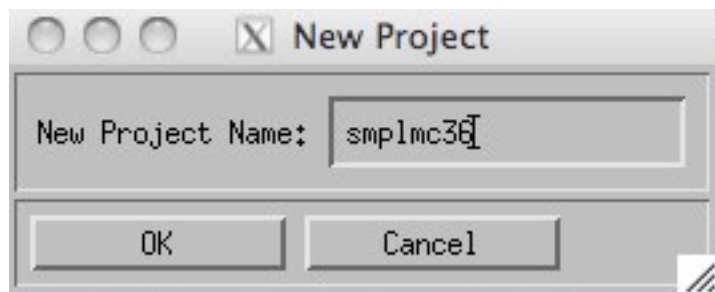
20.1 Start SMART

Type `smart` in the terminal.

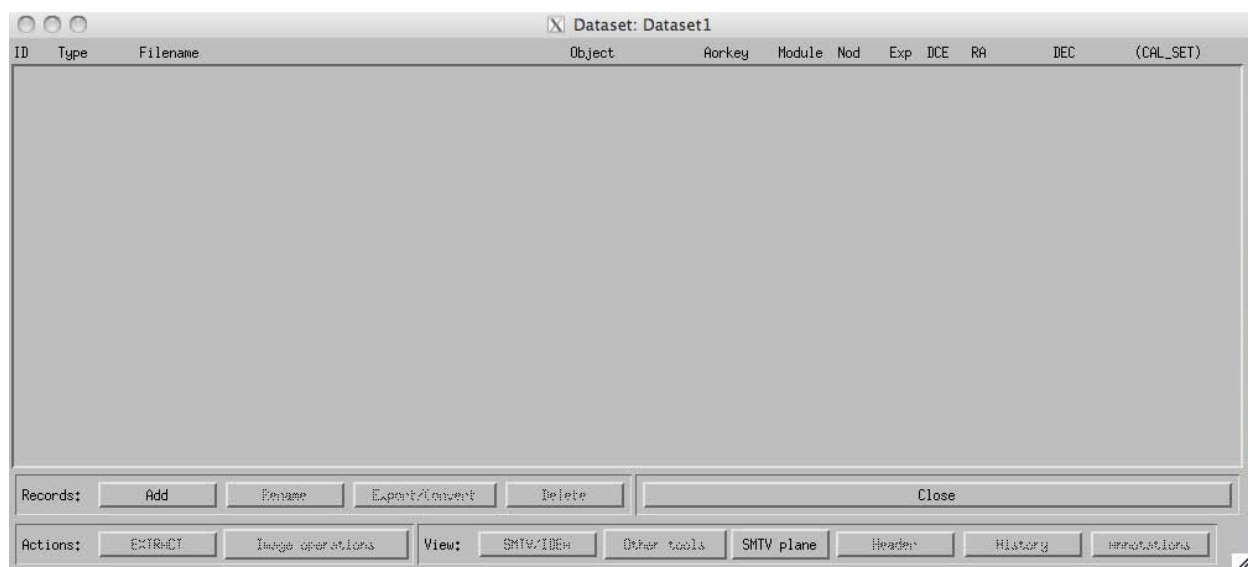
20.2 Data preparation

In the initial GUI create a new project and new data set by clicking on **New** under the **Project** menu. In this case we shall call it `smplmc36`. Click **Ok**.

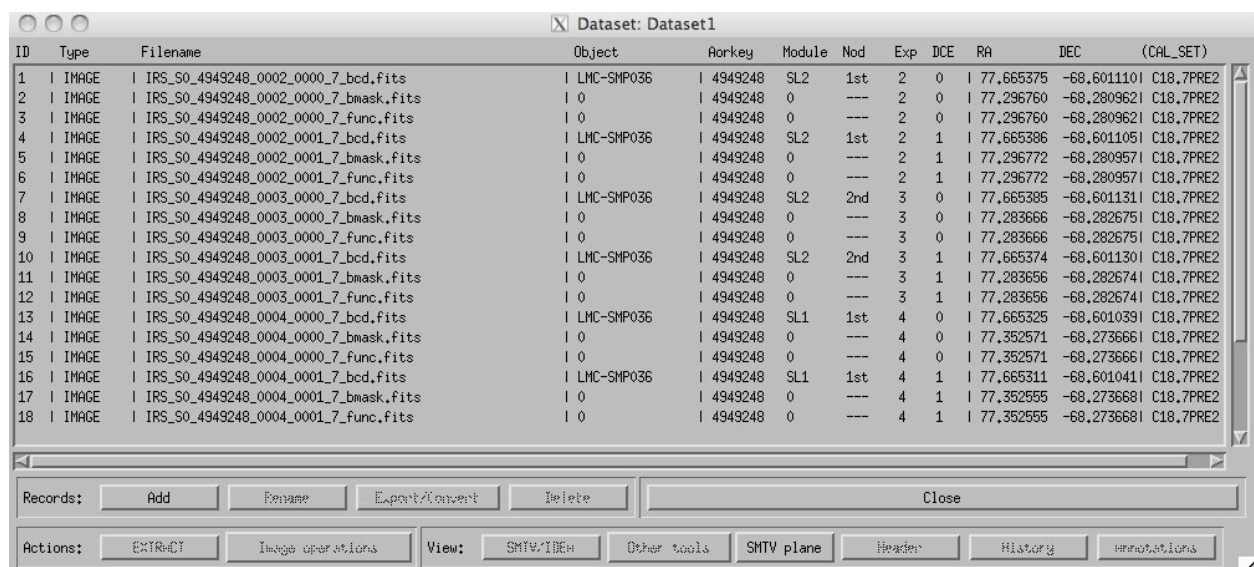
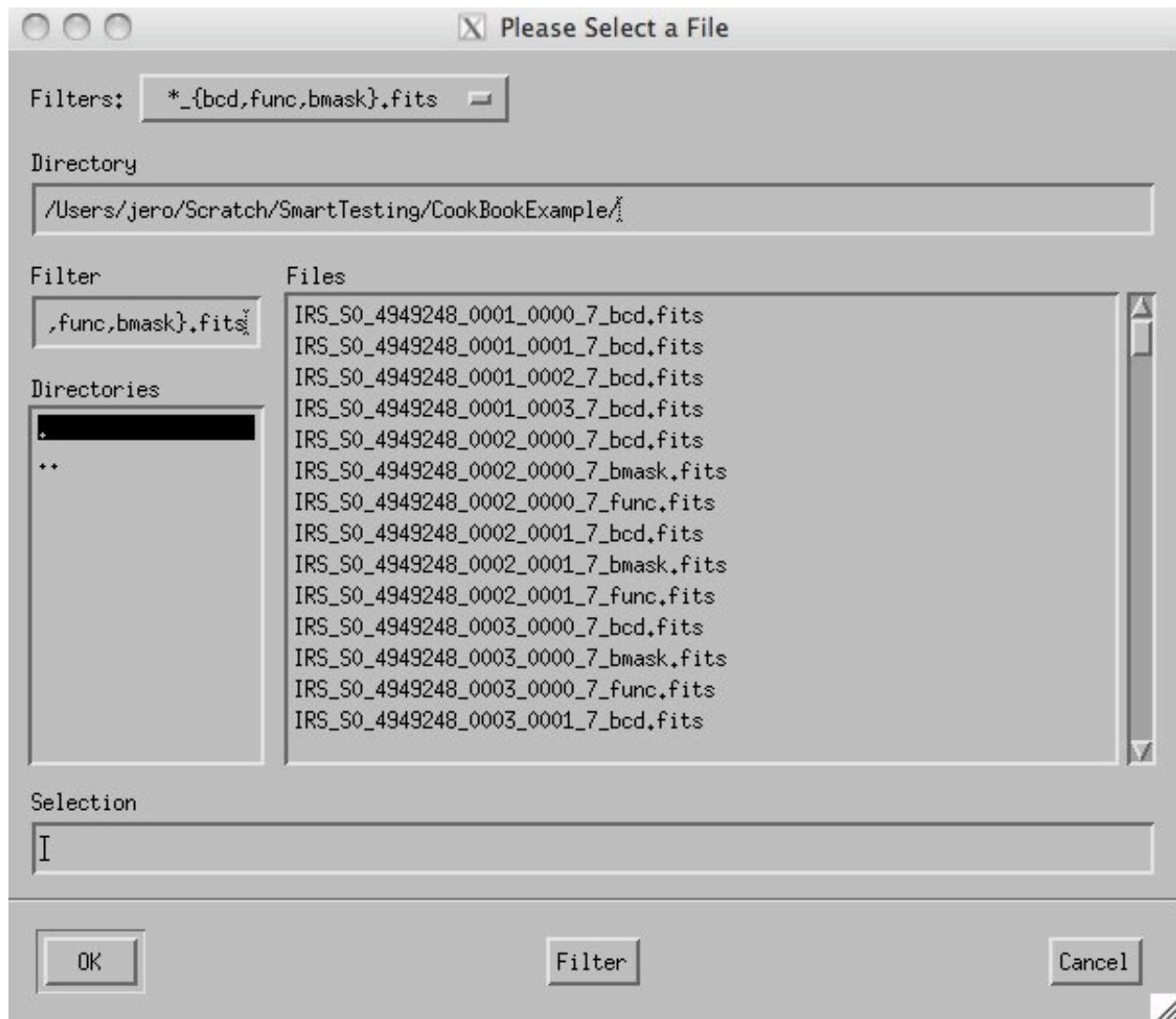




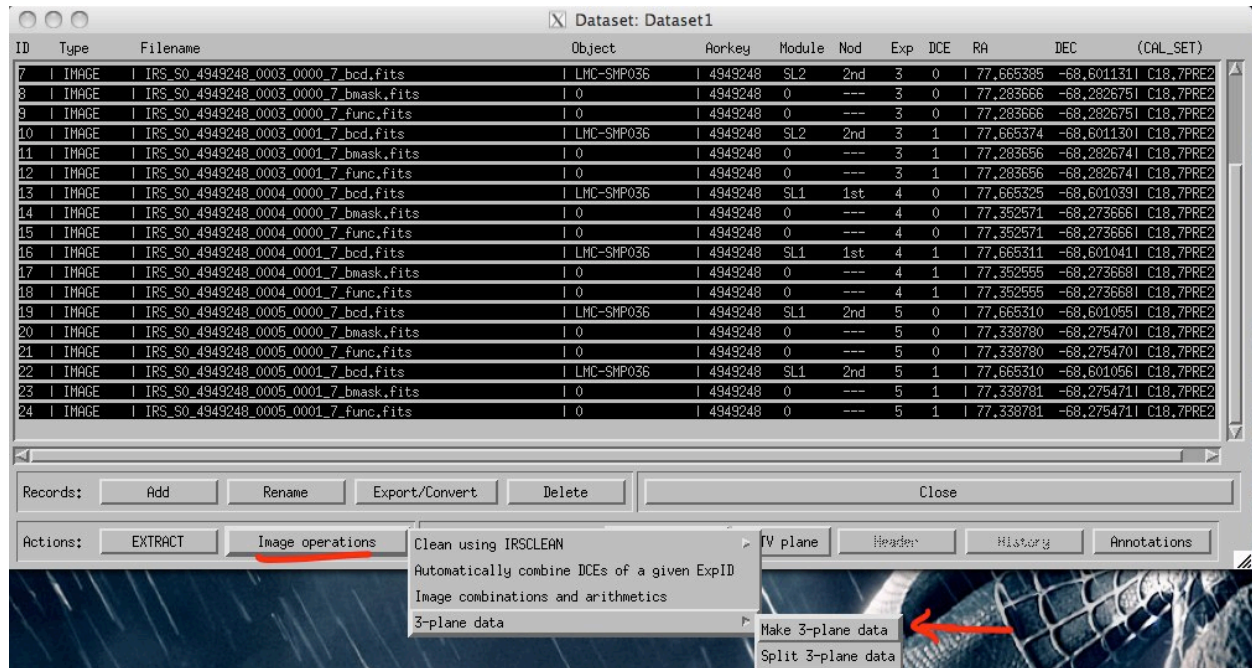
Highlight the *Data set* and either double click on it or click **Edit** button. A new GUI appears.



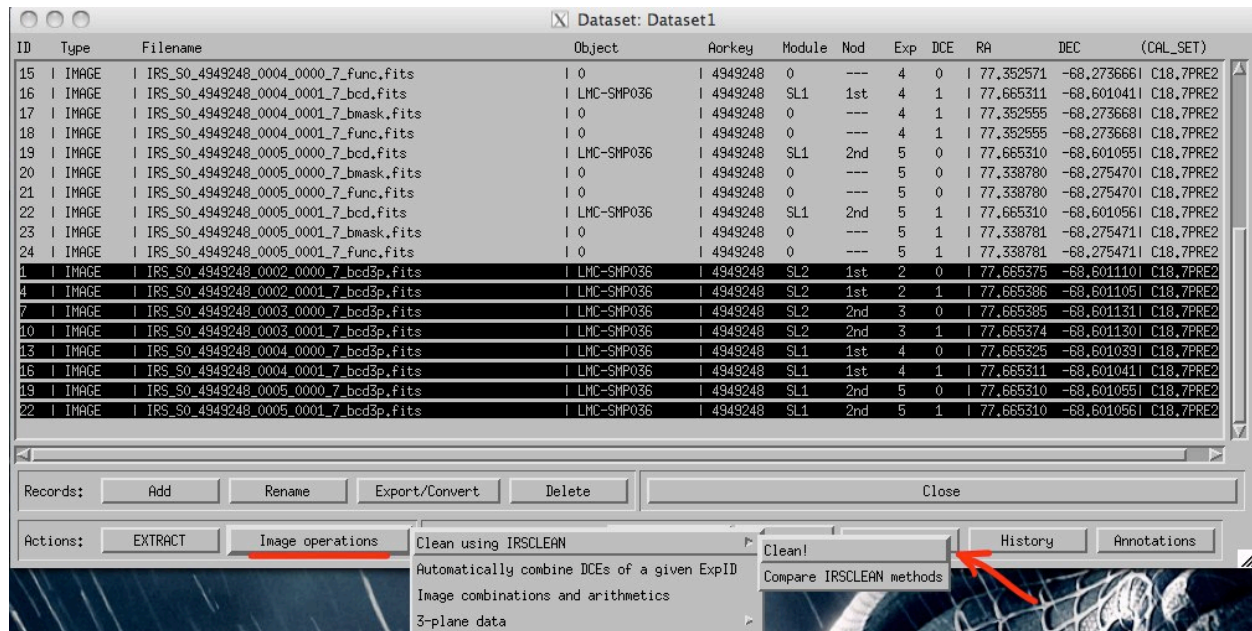
Click on **Add** to add data sets to import the data (`{bcd, func, bmask}.fits`) into the dataset. The new Gui will show the files imported.

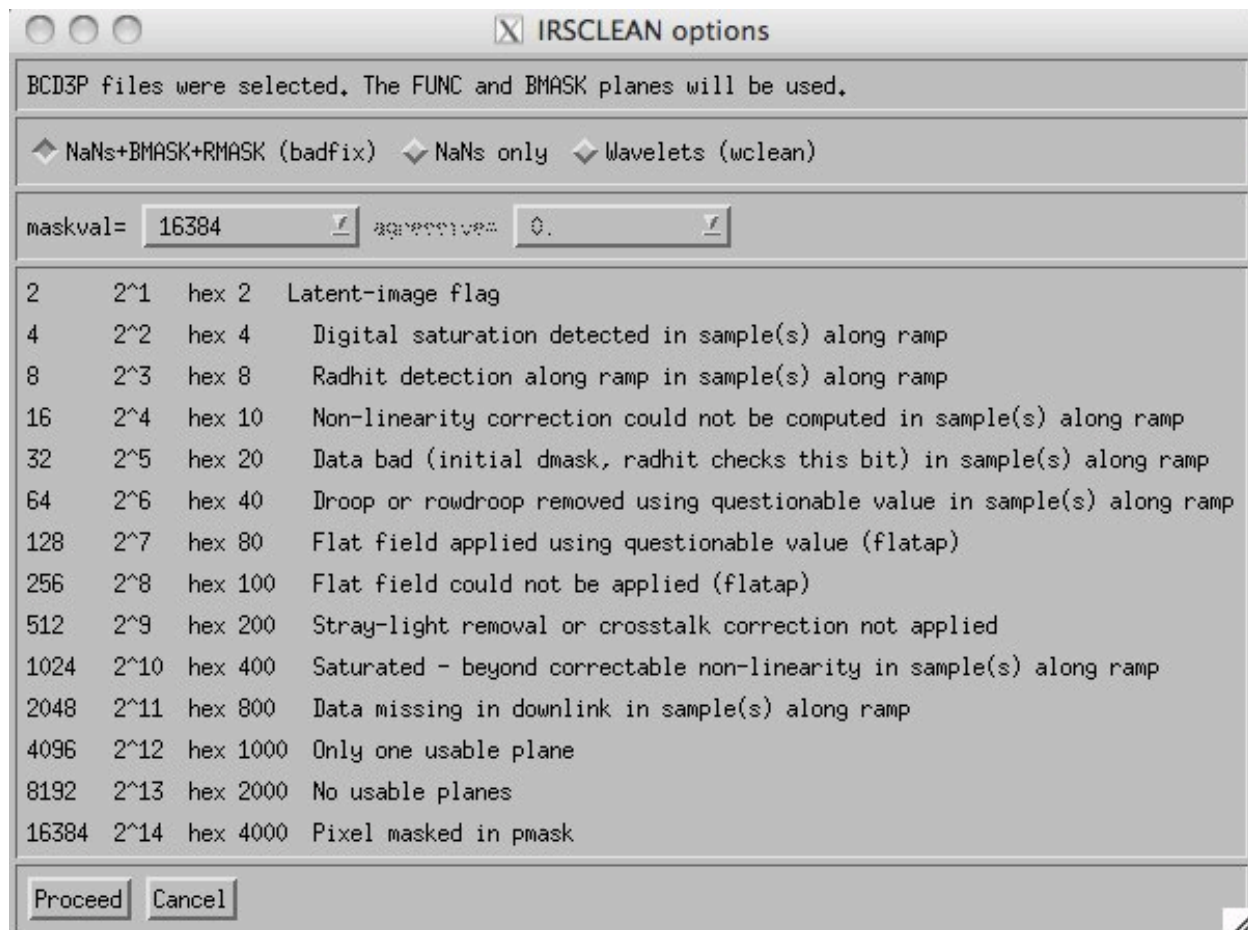


Select all the imported files and click on **Image operations** -> **3-plane data** -> **Make 3-plane data**. This creates *_bcd3p records.

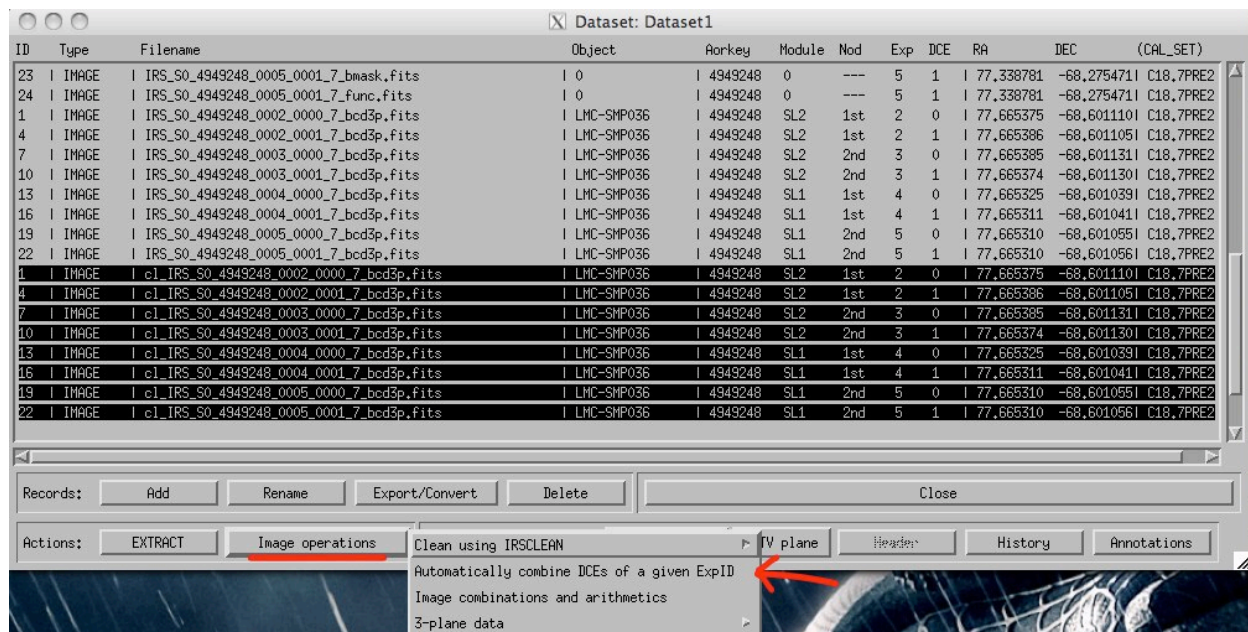


Select the bcd3p records and click on **Image operations** -> **Clean using IRSCLEAN** -> **Clean!**. Leave the default values and click on *Proceed*. This creates cl*_filename.fits records.





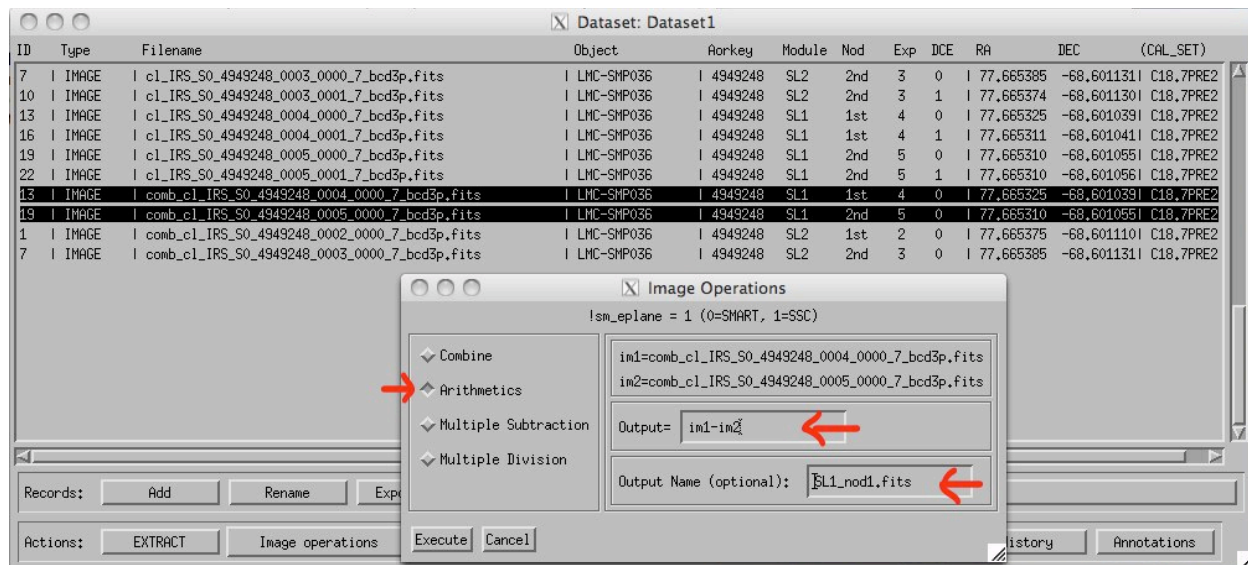
Select all the cleaned records (`cl_*bcd3p`) and click on **Image operations** -> **Automatically combine DCEs for a given ExpID**. This will combine the individual exposures and creates the `com_cl*_filename.fits` records.



20.3 Background subtraction (and low-level rogue pixels removal)

In this example we shall subtract the background from the correspondent nod position.

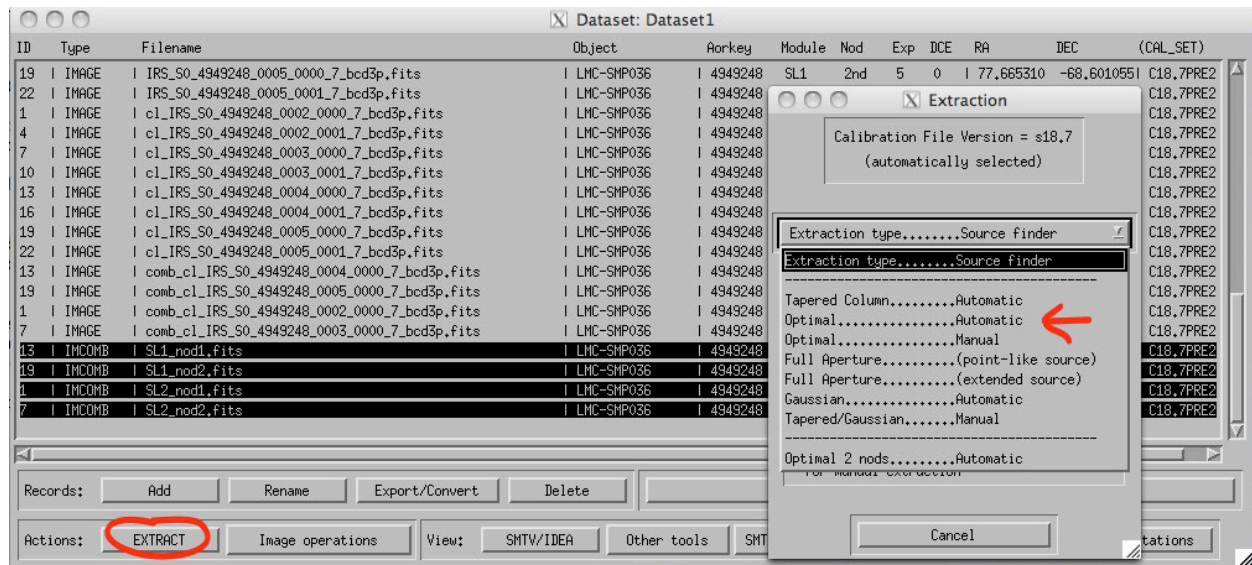
For each module and order (in this case SL1, and SL2), we select the 2 nods and click on **Image operations** -> **Image combinations & arithmetics** -> **Arithmetics**. Then type the appropriate subtraction equation: Input the operation **im1-im2** and call that **SL1_nod1.fits**. Using the same selected files do the operation **im2-im1**, that will contain **SL1_nod2.fits**. Repeat for SL1.



20.4 Spectral extraction

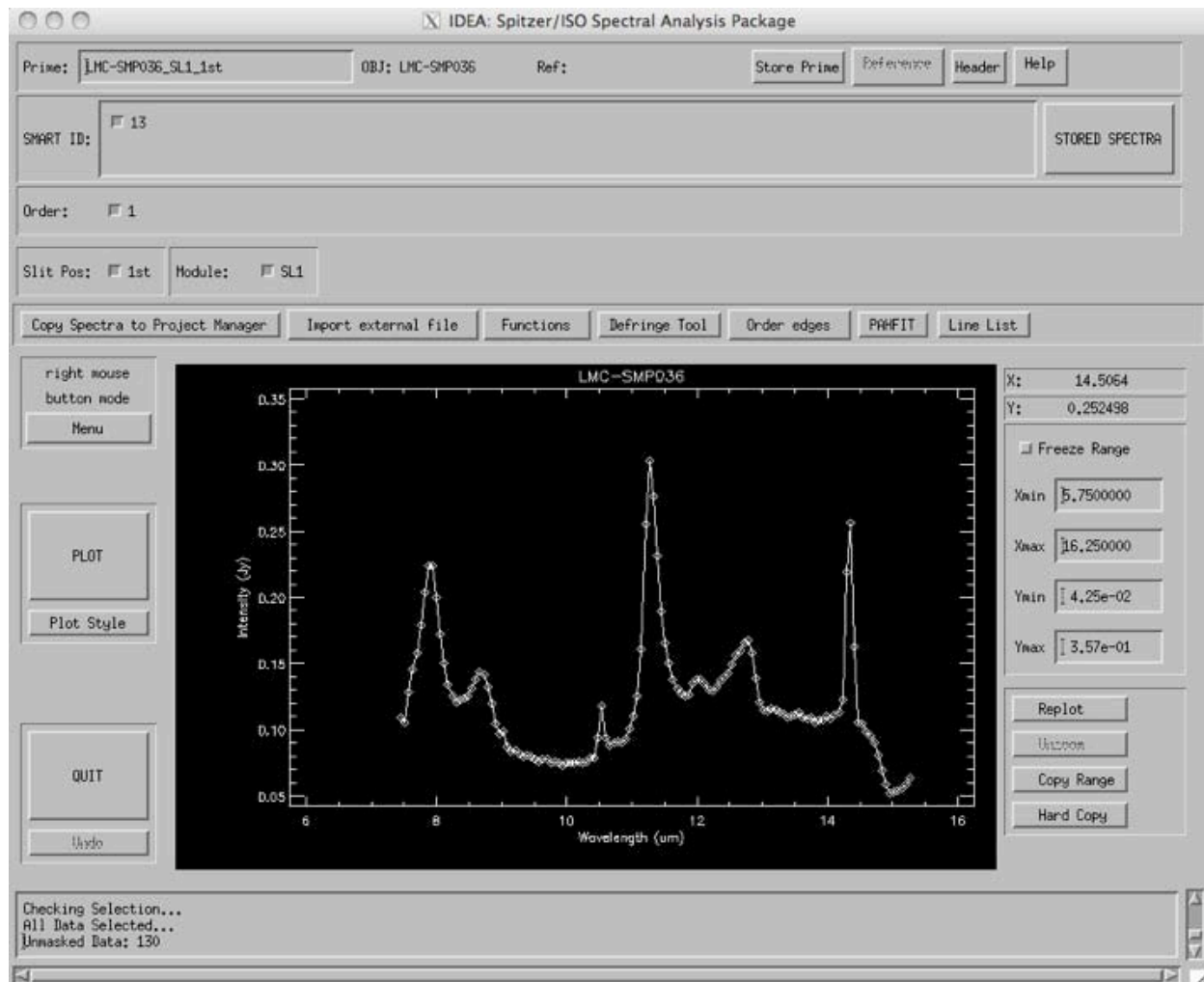
In this case this is a point like source and we are going to performed an automatic optimal extraction. While not shown here, it is advised to run the [AdOpt](#) algorithm on at least one exposure of each module in order to check the actual extent of the source, the possible presence of multiple sources, and/or the presence of a significant extended background emission.

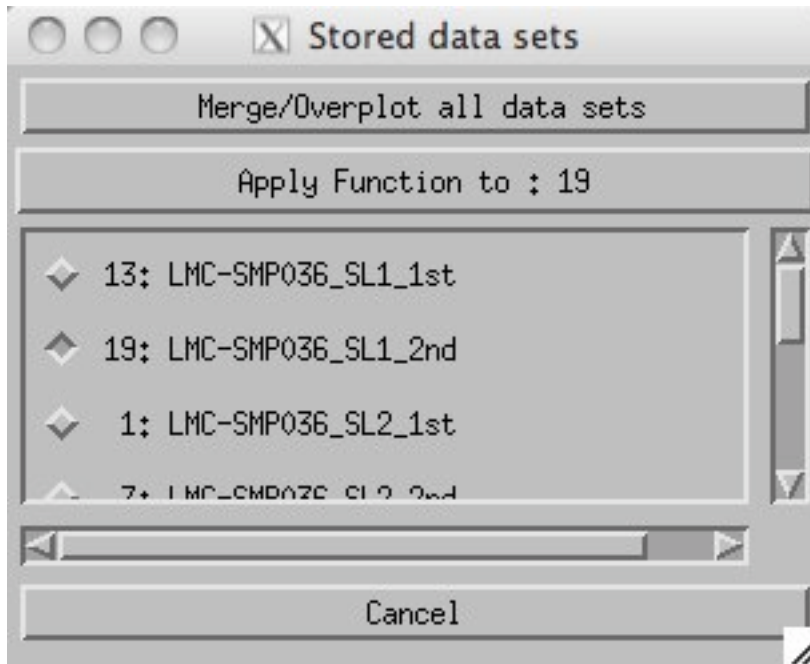
Select all the data. And click on **Extract**. As mentioned before in this case we select **automatic optimal extraction**. One could select automatic tapered or any other option.



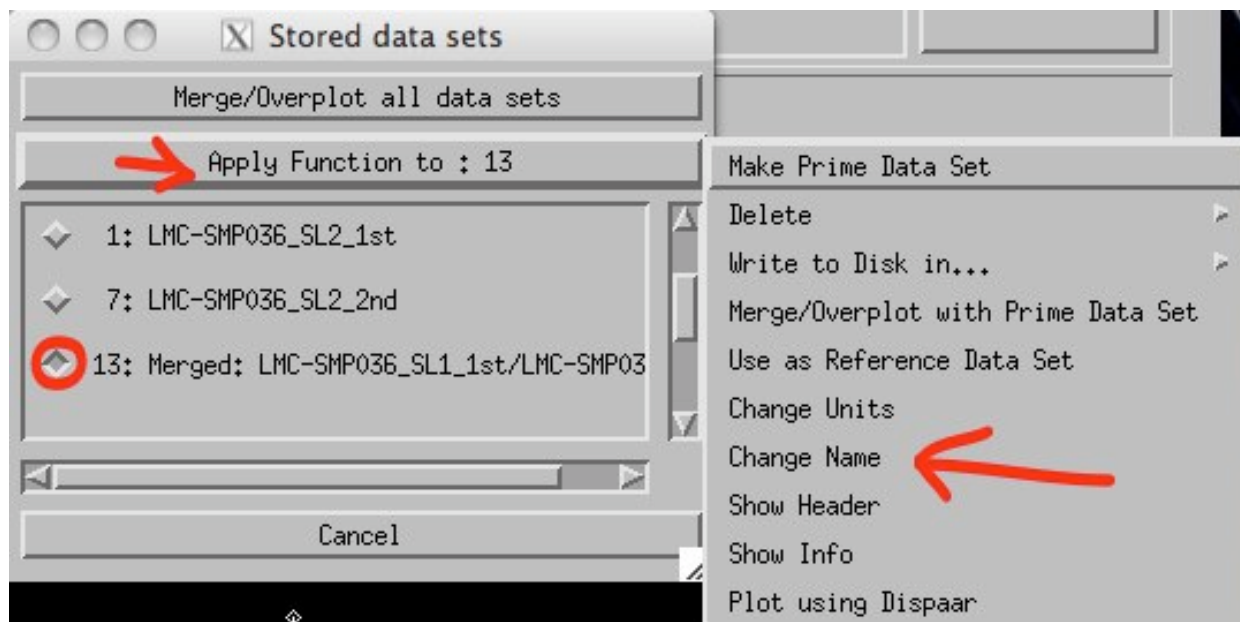
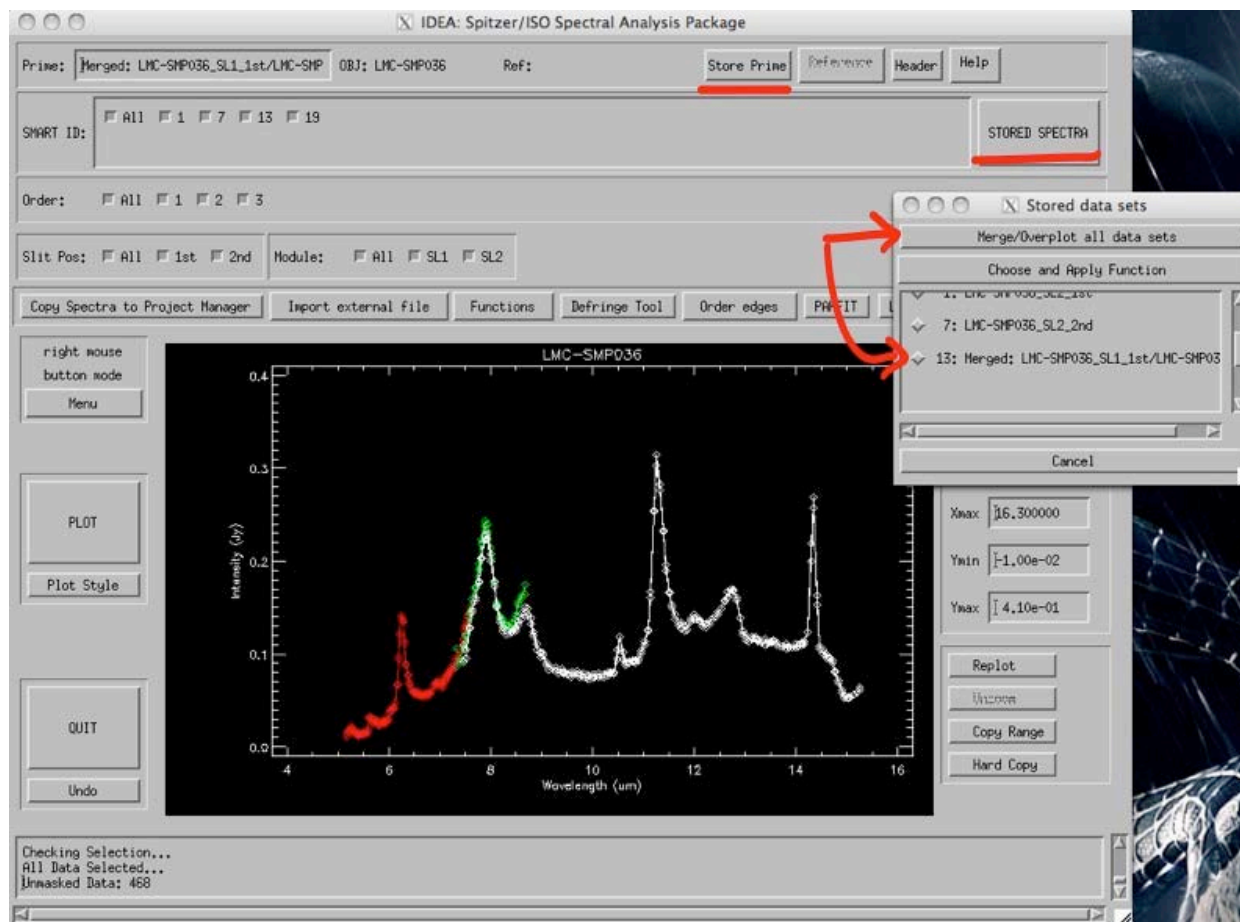
20.5 Spectra visualization and analysis

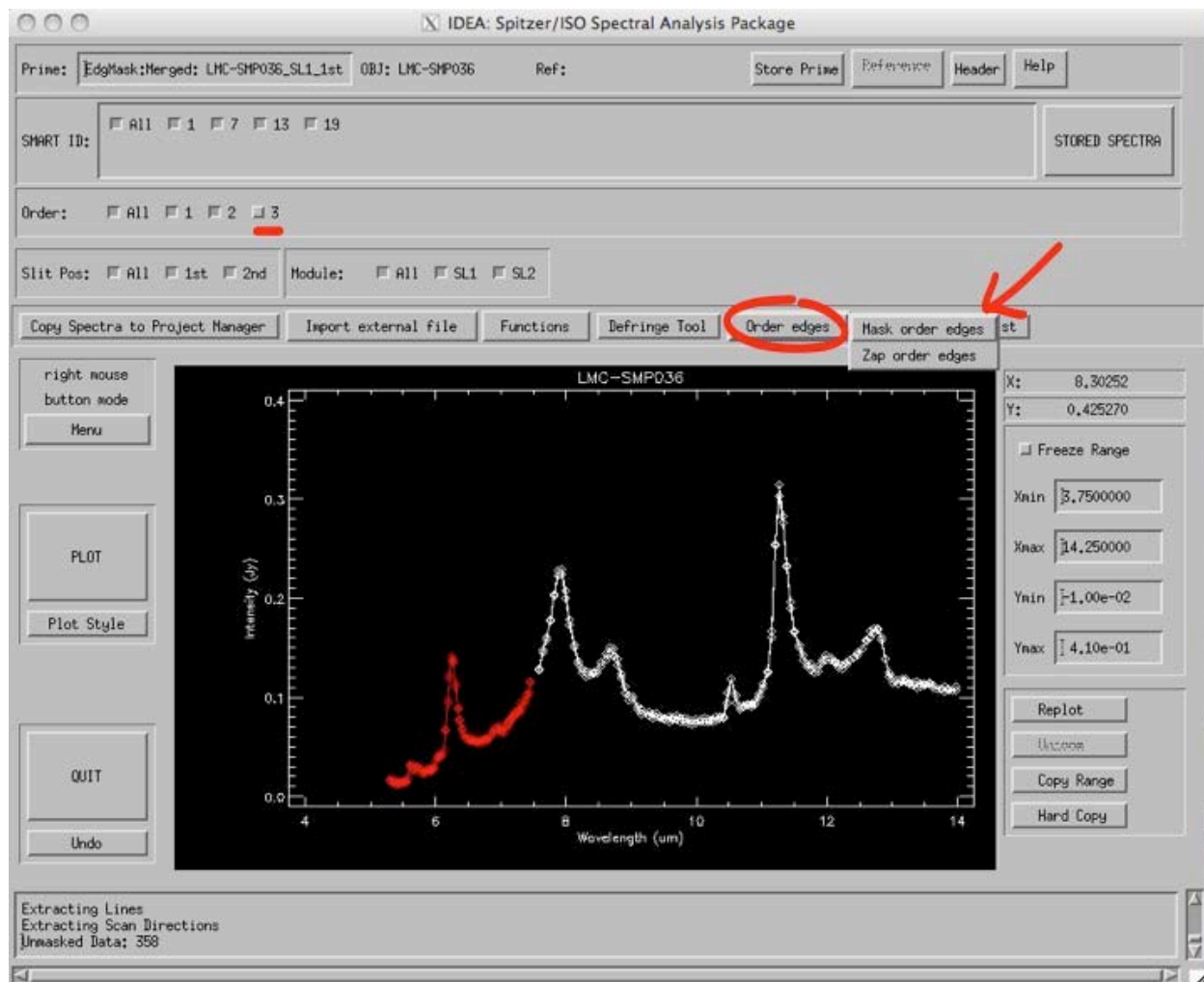
Once the extraction is completed, the spectra is displayed in IDEA automatically. By default only the first spectrum is shown. To display another spectrum, click on the corresponding checkbox in the **Stored datasets** window, click on **Choose an apply a function**, and click on **Make prime dataset**.





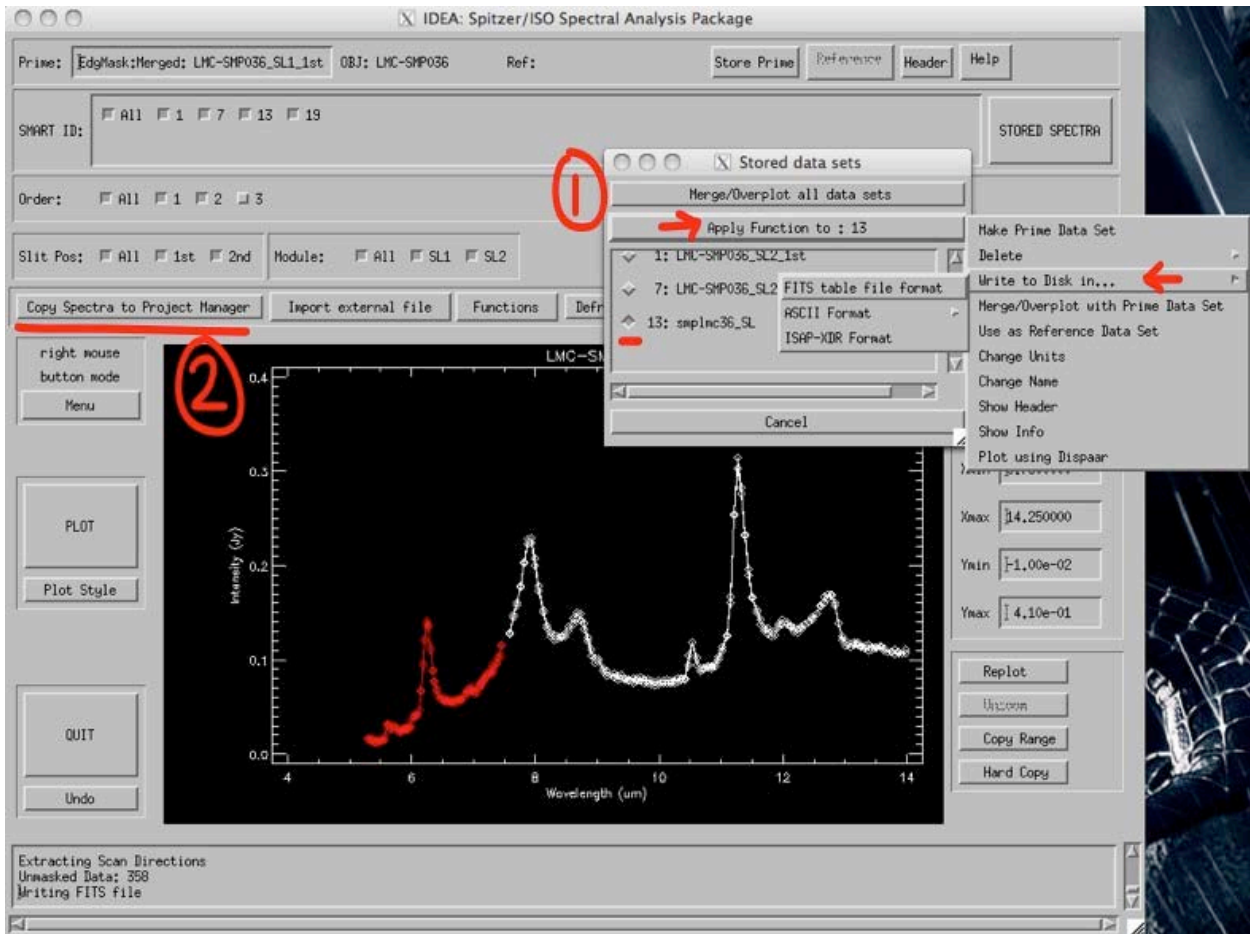
To overplot all the spectra, click on **Merge/overplot all data sets** (shown in figure). In order to store the file containing the merged modules/orders click on **Merge/Overplot all data sets**. Click on **Store Prime** (the Store data sets disappears). Then click on **Stored Spectra** (the updated Store data sets appears). The name of this file is long but we can modify it in the **Store data sets** window by choosing **Apply function*->*Change Name**.





20.6 Saving the data

Click the **Copy Spectra to Project Manager** as shown in the figure (option2) so that the spectra is saved in the project manager. There are several options to store the data in a given directory and format. The data can be written in a given directory (and format) using the store button in the IDEA GUI (option 1 first figure below). Also, the data can be exported form the **Project Manager** GUI as in the second figure below.



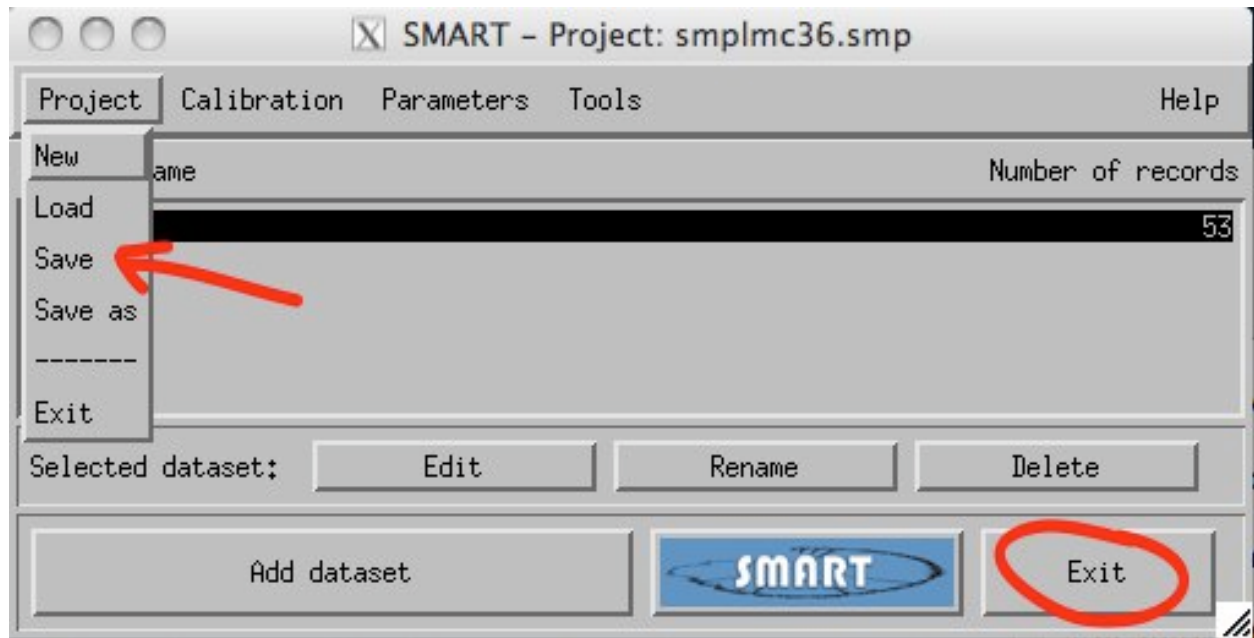
Dataset: Dataset1

ID	Type	Filename	Object	Aorkey	Module	Nod	Exp	DCE	RA	DEC	(CAL_SET)
10	IMAGE	l cl_IRS_S0_4949248_0003_0001_7_bcd3p.fits	LMC-SMP036	4949248	SL2	2nd	3	1	77.665374	-68.6011301	C18.7PRE2
13	IMAGE	l cl_IRS_S0_4949248_0004_0000_7_bcd3p.fits	LMC-SMP036	4949248	SL1	1st	4	0	77.665325	-68.6010391	C18.7PRE2
16	IMAGE	l cl_IRS_S0_4949248_0004_0001_7_bcd3p.fits	LMC-SMP036	4949248	SL1	1st	4	1	77.665311	-68.6010411	C18.7PRE2
19	IMAGE	l cl_IRS_S0_4949248_0005_0000_7_bcd3p.fits	LMC-SMP036	4949248	SL1	2nd	5	0	77.665310	-68.6010551	C18.7PRE2
22	IMAGE	l cl_IRS_S0_4949248_0005_0001_7_bcd3p.fits	LMC-SMP036	4949248	SL1	2nd	5	1	77.665310	-68.6010561	C18.7PRE2
13	IMAGE	l comb_cl_IRS_S0_4949248_0004_0000_7_bcd3p.fits	LMC-SMP036	4949248	SL1	1st	4	0	77.665325	-68.6010391	C18.7PRE2
19	IMAGE	l comb_cl_IRS_S0_4949248_0005_0000_7_bcd3p.fits	LMC-SMP036	4949248	SL1	2nd	5	0	77.665310	-68.6010551	C18.7PRE2
1	IMAGE	l comb_cl_IRS_S0_4949248_0002_0000_7_bcd3p.fits	LMC-SMP036	4949248	SL2	1st	2	0	77.665375	-68.6011101	C18.7PRE2
7	IMAGE	l comb_cl_IRS_S0_4949248_0003_0000_7_bcd3p.fits	LMC-SMP036	4949248	SL2	2nd	3	0	77.665385	-68.6011311	C18.7PRE2
13	IMCOMB	l SL1_nod1.fits	LMC-SMP036	4949248	SL1	1st	4	0	77.665325	-68.6010391	C18.7PRE2
19	IMCOMB	l SL1_nod2.fits	LMC-SMP036	4949248	SL1	2nd	5	0	77.665310	-68.6010551	C18.7PRE2
1	IMCOMB	l SL2_nod1.fits	LMC-SMP036	4949248	SL2	1st	2	0	77.665375	-68.6011101	C18.7PRE2
7	IMCOMB	l SL2_nod2.fits	LMC-SMP036	4949248	SL2	2nd	3	0	77.665385	-68.6011311	C18.7PRE2
13	SPECTRUM	l LMC-SMP036_SL1_1st	LMC-SMP036	4949248	SL1	1st	4	0	77.665118	-68.6010651	C18.7PRE2
19	SPECTRUM	l LMC-SMP036_SL1_2nd	LMC-SMP036	4949248	SL1	2nd	5	0	77.665295	-68.6010661	C18.7PRE2
1	SPECTRUM	l LMC-SMP036_SL2_1st	LMC-SMP036	4949248	SL2	1st	2	0	77.665241	-68.6011271	C18.7PRE2
7	SPECTRUM	l LMC-SMP036_SL2_2nd	LMC-SMP036	4949248	SL2	2nd	3	0	77.665045	-68.6011771	C18.7PRE2
13	SPECTRUM	l smplnc36_SL	LMC-SMP036	4949248	SL1	1st	4	0	77.665118	-68.6010651	C18.7PRE2

Records: Add Rename Export/Convert Delete Close

Actions: EXTRACT Image operations View: SMTV/IDEA Other tools SMTV plane Header History Annotations

*Finally and very important, save the project! and exit.



Recipe 21. MOPEX: Mosaic of MIPS data for HII 173

In this recipe, we will use MOPEX to create a mosaic of the MIPS small-field 24 microns photometry observation of the Pleiades star HII 173. An APEX source extraction recipe follows that uses the same dataset.

21.1 Requirements

- Use the Spitzer archive to download the BCD and post-BCD for AOR 5315584. Recipe 1 has instructions for using the Spitzer archive. We will just be using the 24 microns data.
- MOPEX (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/mopex/>) should be installed as well.

21.1.1 You will need to use this namelist:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/mosaic_24um.nl

21.2 Step-by-Step Guide

1. The online (=automatically produced) 24 microns mosaics are very good, often good enough to start analyzing right away, depending on your science. For an initial look at your data, use ds9 or your favorite image viewer to examine the on-line mosaics that came with your data.

```
unix% cd /where/you/unpacked_data/r5315584/ch1/pbcd/  
unix% ds9 SPITZER_M1_5315584_0000_1_E429953_msaic.fits &
```

Note that there are two mosaics in this directory. The first is the one where the 24 microns array is "prime", and the other is where it is not, e.g., the ancillary data. Those non-prime data are *not* what we are interested in here. There are lots of sources here!

2. Set up things to run, part 1.

Personally, I like to have just one copy of the *cal/* directory into which I can put the latest pmasks and the PRF I want to start from (check the SSC web site for the latest versions of these files:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/mips/calibrationfiles/>). Also copy the cdf namelist files for this example into a *cdf/* subdirectory.

```
unix% cd r5315584/ch1/bcd/  
unix% ln -s /where/you/unpacked_data/cal ./cal  
unix% mkdir cdf  
unix% mv /where/you/downloaded/mosaic_24um.nl ./cdf/
```

3. Set up things to run, part 2.

```
unix% cd r5315584/ch1/bcd/  
unix% ls *bcd.fits > InputImageList.txt  
unix% ls *bmask.fits > DmaskList.txt  
unix% ls *bunc.fits > SigmaList.txt  
unix% nedit InputImageList.txt DmaskList.txt SigmaList.txt  
unix% wc *.txt  
unix% nedit cdf/mosaic_24um.nl
```

The **.txt* files are needed as input to the *mosaic.pl* script. These lists will contain all of the bcds, masks, and uncertainty files.

You don't want to list every single file here, because (i) there are serendipitous data (e.g., data obtained using the 24 microns array when the spacecraft and scan mirror motions are optimized for the 70 and 160 microns arrays), and (ii) the first frame effect means that you want to drop the DCENUM=0 frames. (Note: if you really want to keep the first frames, those data can be corrected and included, but we have plenty of data, so we have the luxury of simply dropping them. See the MIPS Instrument Handbook for more details on this.) Delete the files from the lists starting with the following roots to remove the first frames:

```
SPITZER_M1_5315584_0000_0000_1_*  
SPITZER_M1_5315584_0001_0000_1_*  
SPITZER_M1_5315584_0002_0000_1_*
```

To figure out which files are serendipitous data, the easiest way to do this is to look in the header for the keyword PRIMEARR. You can examine each header separately, but this is how I do it in a more automated fashion. I use the tool "imhead" (available from the unix command line), which is part of the WCS tools package (<http://tdc-www.harvard.edu/software/wcstools/index.html>). Type "which imhead" at the unix command line to see if you have that package installed (if it says "command not found," you don't have it) and go install it if necessary.

```
unix% which imhead  
/SciApps/bin/imhead  
unix% imhead *bcd.fits | grep PRIMEARR > aaa  
unix% ls *bcd.fits > bbb  
unix% paste aaa bbb > ccc
```

The file "ccc" then has a listing of the PRIMEARR keyword for each file followed by the corresponding filename. In this fashion, you can figure out which files are prime and which are not. For this set, the files starting with the following roots are the serendipitous data. Edit the **.txt* files correspondingly to remove them:

```
SPITZER_M1_5315584_0003_*  
SPITZER_M1_5315584_0004_*  
SPITZER_M1_5315584_0005_*
```

Then check the length of each file ("wc *.txt") to be sure that you deleted the same number of files from each list. The "wc" step counts the number of lines, words, and characters in the files; the total number of lines should be the same in each of the *.txt files. If they're not, fix it so that they are...

Finally, edit the namelist to reflect where your cal files are located -- be sure to use the right pathnames!

4. Make the mosaic!

```
unix% mosaic.pl -n mosaic_24um.nl
OR
unix% mkdir pbcd24
unix% mosaic.pl -n mosaic_24um.nl > pbcd24/mosaic.log
```

Note that the namelist is *assumed* to be in the *cdf/* subdirectory, not in your current directory. If desired, redirect the output to a log file for your records. (actual mosaic log file:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/mosaic24.log>)

Common errors:

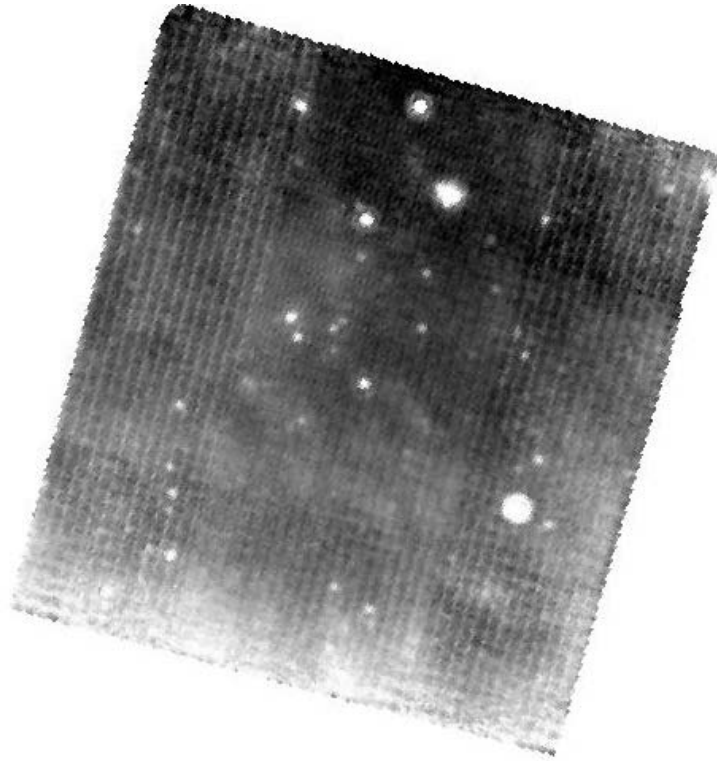
- "mosaic.pl: Command not found." means you have not sourced the csh when setting up the mopex package in your current shell (window).
- other "files not found" -- make sure you set all the paths correctly in the namelist, and that you're sitting in the correct directory when you call *mosaic.pl*.

5. Check the mosaic.

```
unix% ds9 pbcd24/Combine/mosaic.fits &
```

Check the mosaic for anything that looks 'odd.' It should look pretty much like the online mosaic (which came with your data, in the pbcd directory). See figure below and FITS file

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/my24mosaic.fits>.



Common errors:

- If you didn't remove all of the ancillary data, you'll wind up with what looks like two separate mosaic pieces in the same FITS file. Go back and be sure you removed all the ancillary data.
- If you didn't remove the first frames, you'll get sharp gradients across the frame; see the MIPS Instrument Handbook. A slow gradient across is fine, and may even be real zodiacal light variations.

Recipe 22. MOPEX/APEX: MIPS 24um mosaic of a $z=0.7$ cluster

For each MIPS observation, three types of files are available for download from the Spitzer archive: Raw, BCD (Basic Calibrated Data), and post-BCD. Each represents a different stage of processing. As explained in the MIPS Instrument Handbook, the pipeline that produces the BCD products is appropriate for most science programs. However, you may find that the post-BCD pipeline does not remove all instrumental artifacts (e.g. jailbars). If this is the case, then you may wish to start with the BCD products and carry out any further processing on your own. This tutorial provides an example for a MIPS staring-mode observation (a $z=0.7$ galaxy cluster), describing how to recognize artifacts in your post-BCD data and how to remove these artifacts from your BCD products using *flatfield.pl*, a perl script shipped with MOPEX. The tutorial then goes on to show you how to mosaic your corrected BCD frames using MOPEX, and how to perform source extraction using APEX, which is distributed with MOPEX.

22.1 Requirements

You must install MOPEX

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/mopex/>) in order to follow along with this recipe.

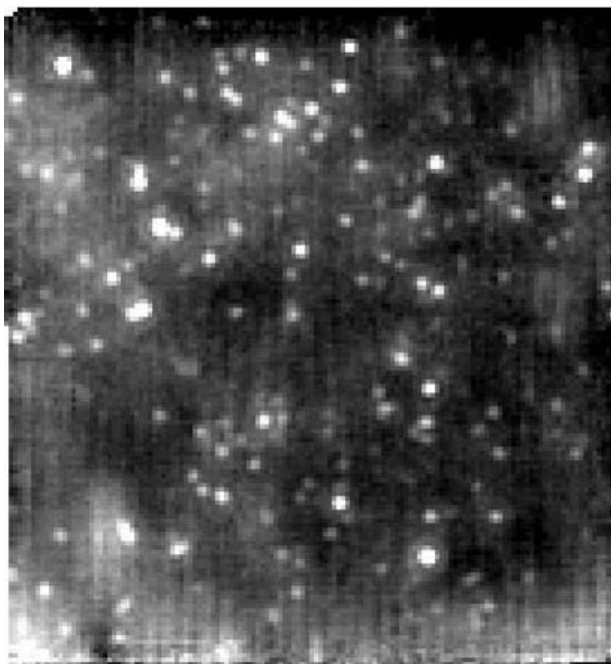
You will need 281 Mb disk space to hold the downloaded data. Download data from the Spitzer archive (see Recipe 1): BCD, post-BCD, and calibration data associated with AOR 13810432.

If you run MOPEX and wish to keep all the intermediate data files, this will require another 400 Mb of disk space. If you do not wish to keep the intermediate data files (only possible if you are using the command-line version of MOPEX; see below), this will be significantly reduced, to about 50 Mb.

22.2 Recognizing artifacts in the Post-BCD data

Let's take a look at the post-BCD data to get a sense of what our science data look like. In the directory *r13811456/*, there is only one subdirectory, called "*ch1/*". This is the directory where the 24 microns data are stored (the 24 microns data are channel 1, while the 70 and 160 microns data are channels 2 and 3, respectively). Within the *ch1/* directory are three subdirectories corresponding to the data types you selected for download: "*bcd/*", "*cal/*", and "*pbcd/*". These hold the BCD, calibration, and post-BCD data products, respectively.

For a quick look at the data, cd into the *pbcd/* directory. There is one file in this directory that ends in "*maic.fits*". This is the mosaic image made from the individual pointings. Below is a JPEG of this mosaic:



SPITZER_M1_13810432_0000_3_E3189129_maic.fits

The JPEG above shows the attempt made by the post-BCD pipeline to mosaic together the BCDs for this pointing. As mentioned above, the post-BCD pipeline may not be optimized for every science program. Indeed the image above contains instrumental artifacts that we will attempt to remove. A full list of these, along with methods for mitigating their effects, can be found in the MIPS Instrument Handbook. Particularly noticeable are the jailbars, which appear as vertical stripes. Also present are bright and dark latents, which make the background appear splotchy. Since the sources in this field are all rather faint, these latents are likely caused by previous observations of bright sources.

22.3 Self-calibration to remove artifacts

To remove the artifacts seen in the mosaics that come out of the post-BCD pipeline, we first need to create a frame which consists only of the artifacts. We can then use this frame to divide out the artifacts in each BCD image. These steps can be performed with the perl script *flatfield.pl*, which comes with MOPEX, but has not yet been ported to the GUI. For each BCD file, *flatfield.pl* performs background subtraction, source detection on the background-subtracted images, stacking of background-subtracted images to create a flat while masking out detected sources and bad pixels, and flatfielding of the BCD images.

The user can achieve the functionality of *flatfield.pl* with a combination of SExtractor (for the detection step) and IRAF or IDL (for all of the other steps).

The inputs to *flatfield.pl* are:

1. A **namelist file**, which allows the user to set some tunable parameters. For this example, we suggest you use this heavily commented namelist, http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/flatfield_24_ediscs.nl. You should make a directory called *r13810432/ch1/cdf/* and put this file in it.

Note that if you have limited diskspace, you could change the line that currently reads "delete_intermediate_files = 0" to "delete_intermediate_files = 1".

2. A **list of input images, a list of sigma images, and a list of Dmask images**. MIPS 24 microns data suffer from the "First-frame Effect", where the first frame of every commanded sequence of observations (data with the keyword DCENUM=0) has a shorter exposure time and is depressed in response by 10-15%. The MIPS instrument support team recommends discarding this first frame. In addition, the second and even third BCDs (DCENUM=1,2) also appear to have a reduced signal (~2%). If you have enough data then you may also want to consider discarding these. For this example, we suggest you discard the first frames:

```
cd r13810432/ch1
mkdir firstframes
cd bcd
mv SPITZER*_000*_0000*.fits ../firstframes
ls FULLPATH/r13810432/ch1/bcd/SPITZER*bcd.fits >
InputImageList.txt
ls FULLPATH/r13810432/ch1/bcd/SPITZER*bunc.fits > SigmaList.txt
ls FULLPATH/r13810432/ch1/bcd/SPITZER*bbmsk.fits > DmaskList.txt
```

Here, *FULLPATH* is not literal but represents the entire path to your download directory. To run *flatfield.pl*, first place the above **.txt* files in the *r13810432/ch1/* directory. Then:

```
cd r13810432/ch1
csh
source /Applications/mopex/mopex-script-env.csh
flatfield.pl -n flatfield_24_ediscs.nl
```

This should take less than 2 minutes to run to completion. The script will have created a new subdirectory called *mopex_flat/*, since that is the directory specified by the namelist file. By entering the *mopex_flat/* directory, you can examine the flat (named *flat.fits*) created by *flatfield.pl*. With a hard stretch, it should look something like this:



flat.fits

Note that flat consists only of artifacts, since the sources have been masked out. These artifacts include permanently-damaged pixels (which look somewhat like point sources and are additionally masked by MOPEX during mosaic creation), as the bright and dark latents mentioned above. Jailbars are also present, but are at a very low level.

The contents of the subdirectories created by *flatfield.pl* are as follows (if `delete_intermediate_files = 0` in the namelist file):

Medfilter/: This directory contains background images and background-subtracted images corresponding to each of your input BCDs.

Detector/: This directory contains mask files, so that certain pixels in each image will be excluded in the creation of the flatfield image. The masked pixels are associated with sources detected in the background-subtracted input images.

Correct/: This directory contains the flatfielded input images. These are the images you will use to make your new mosaic, as described in the next section.

22.4 Make a new mosaic using your self-calibrated BCDs

The final step is to re-create the post-BCD mosaic using your self-calibrated BCDs. There are two ways you can do this. You can either use the script *mopex.pl* that comes packaged with MOPEX, or you can use the MOPEX GUI. We illustrate both methods below.

To make the mosaic from the command line:

Download this namelist file:

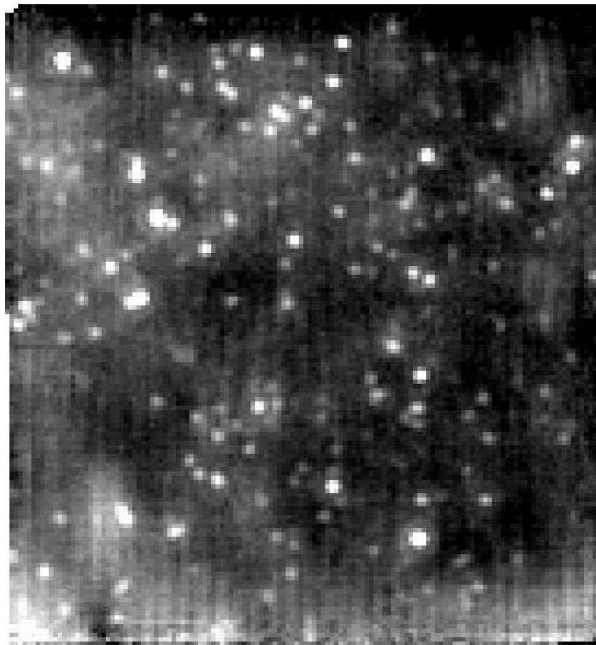
http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisstools/cookbook/files/mosaic_24_ediscs.nl and put it in the directory *r13810432/ch1/cdf/*.

Note that if you have limited disk space, you could change the line that currently reads "delete_intermediate_files = 0" to "delete_intermediate_files = 1".

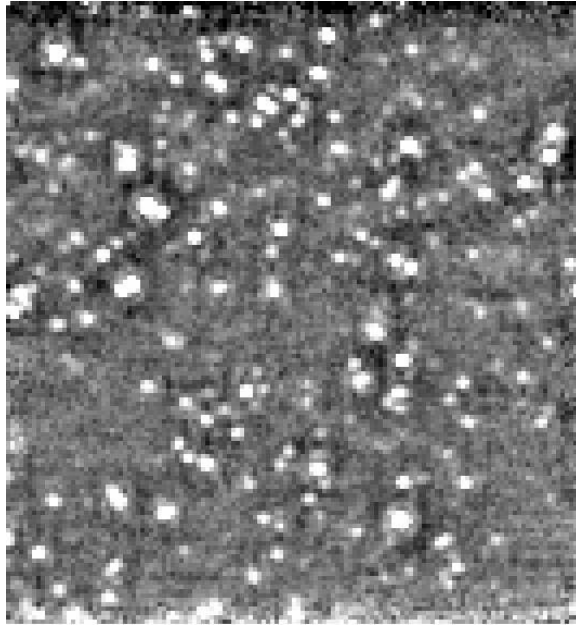
```
cd r13810432/ch1
csh source /Applications/mopex/mopex-script-env.csh
mosaic.pl -n mosaic_24_ediscs.nl
```

While you are waiting for *mosaic.pl* to finish executing, you can read about exactly what is doing in our online documentation for the mosaic outlier.

Once the above command has finished executing, you will notice that a new directory called *pbcd/* has been created. Within this directory are several subdirectories. The most important of these is the *Combine/* subdirectory, which contains the new, self-calibrated mosaic you just made. We show it next to the automatic mosaic created by the post-BCD pipeline without self-calibration, so you can see the improvement.



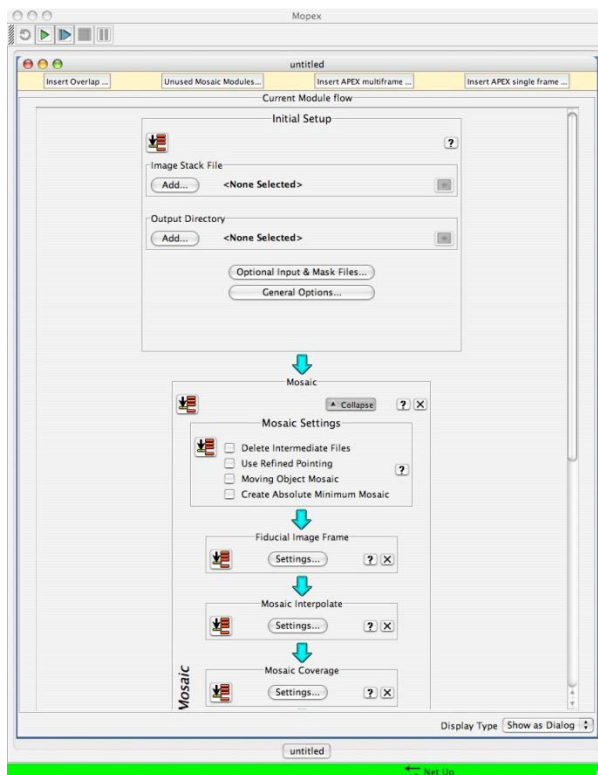
BEFORE: the mosaic created automatically by the post-BCD pipeline,
r13810432/ch1/pbcd/SPITZER_M1_13810432_0000_3_E3189129_maic.fits



AFTER: the mosaic created by you after self-calibrating the BCDs,
r13810432/ch1/pbcd/Combine/mosaic.fits

To use the MOPEX GUI:

Start up the MOPEX GUI. A new window will pop up, but it will be empty. In the main MOPEX menu, click on File --> New Mosaic Pipeline. You will be asked to select a template. Select "Mosaic, MIPS 24 microns". The previously-empty MOPEX window will now be filled, and will look something like this:



Instead of feeding the MOPEX GUI a namelist like the one you used for the command-line version, you can click on various menus in the GUI to control how your data are reduced.

The mosaic pipeline consists of two main sections: "Initial Setup" followed by "Mosaic". Below is a listing of which parameters to set in each.

1. Set the parameters which should be changed from the template defaults:

Image Stack File r13810432/ch1/mopex_flat/Correct/correct_InputImageList.txt

Output Directory To differentiate this run from the command-line run, first create a directory called "postbcd_gui" for the output files to be written to:

```
cd r13810432/ch1
```

```
mkdir postbcd_gui
```

Then, in the MOPEX GUI, select

```
r13810432/ch1/postbcd_gui
```

Optional Input & Mask Files:

Sigma List File SigmaList.txt

DCE Status Mask List DmaskList.txt

and leave the "Fatal Mask Bit Pattern" at the template default

Pmask FITS File cal/mips24_pmask.fits

and leave the "Fatal Mask Bit Pattern" at the template default

2. Run MOPEX

At the top of the MOPEX screen you will see a green right-pointing arrow:



Click on it to run MOPEX. You will be able to watch as MOPEX progresses through the flow.

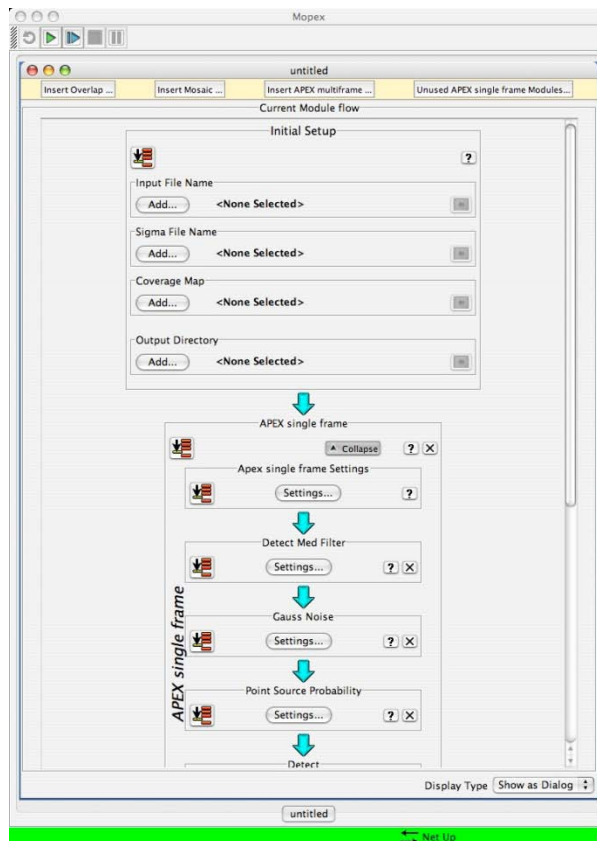
22.5 Perform source extraction on your new mosaic

Now we will demonstrate how to use the post-BCD tool APEX to produce a catalog of sources in the mosaic you made in the previous steps.

APEX has two pipelines: single-frame point source extraction and multi-frame point source extraction.

In this example, we will run APEX in single-frame mode, using the mosaic we created using the MOPEX GUI, and the 24 microns PRF provided with MOPEX. The basic steps are outlined below:

1. Start a new APEX single-frame pipeline. After starting up MOPEX, go to the main menu and select File-->New Apex Single-Frame Pipeline. In the window that pops up, select "APEX 1frame, MIPS 24 microns". The previously-empty MOPEX window will now be filled with a module flow consisting of "Initial Setup" and "APEX single frame". It will look like this:



2. Set parameters.

Only the parameters that require a change from the template default are listed below.

Initial Settings:

Input File Name r13810432/ch1/pbcd/Combine/mosaic.fits
Sigma File Name r13810432/ch1/pbcd/Combine/mosaic_unc.fits
Coverage Map r13810432/ch1/pbcd/Combine/mosaic_cov.fits
Output Directory r13810432/ch1/apex_output

APEX single frame settings:

PRF File Name /Applications/mopex/cal/mips24_prf_mosaic_2.45_4x.fits

You will have to modify your path to reflect your mopex installation directory

Aperture Photometry:

Number of Apertures Five

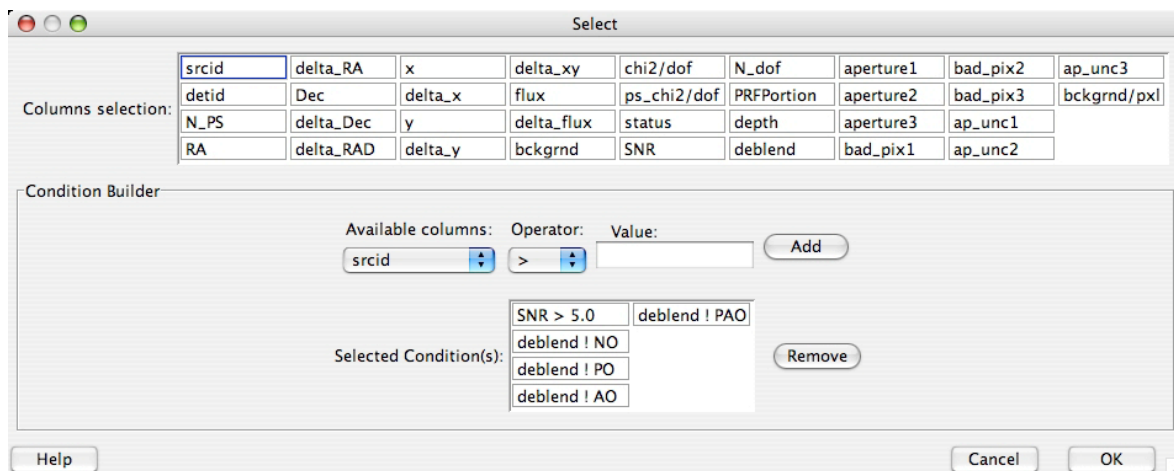
Number of circular apertures to draw around each source, between 1 and 5.

Aperture Radius 1, 2, 3 1.0, 1.5, 2.0, 2.6, 3.0

Radius of aperture N in pixels.

Select:

This module creates the final table by applying user-specified constraints and copying user-specified columns. The Settings window in the MOPEX GUI looks like this:



Columns Selection

Click on the columns you would like to appear in your final output table. On macs, use the COMMAND key to select more than one column. On PCs and machines running Solaris, use CNTRL.

Condition Builder

You can retain only certain objects in your catalog, based on their measured properties. For instance, you may wish to keep only sources with signal-to-noise ratios greater than 2. In that case, you would first remove the template condition SNR > 5. To do this, go to the "Selected Condition(s)" section of the GUI, and click on "SNR > 5.0". Then click on "Remove". To add your new condition, set the following options just above:

Available Columns = SNR
Operator = >
Value = 2

And then click "Add". Under "Selected Condition(s):", your new condition will appear as "SNR > 2.0".

For a complete list of all of the parameters associated with this module, open the "Help" window from the main MOPEX menu, and then scroll down the menu on the left-hand side to choose "Select". Alternatively, you can close the Settings window and click on the question mark that appears next to the Settings button.

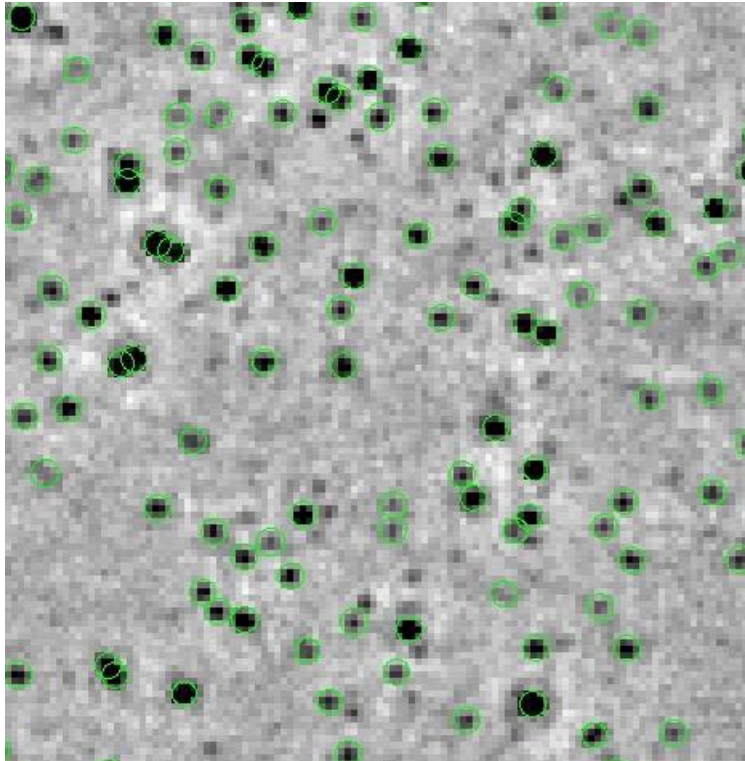
3. Run APEX. Click on the green arrow at the top of the MOPEX window. When it has finished running (it should be very quick), you will find your output photometry table in the directory:

r13810432/ch1/apex_output/

The final extraction table is called

mosaic_extract.tbl

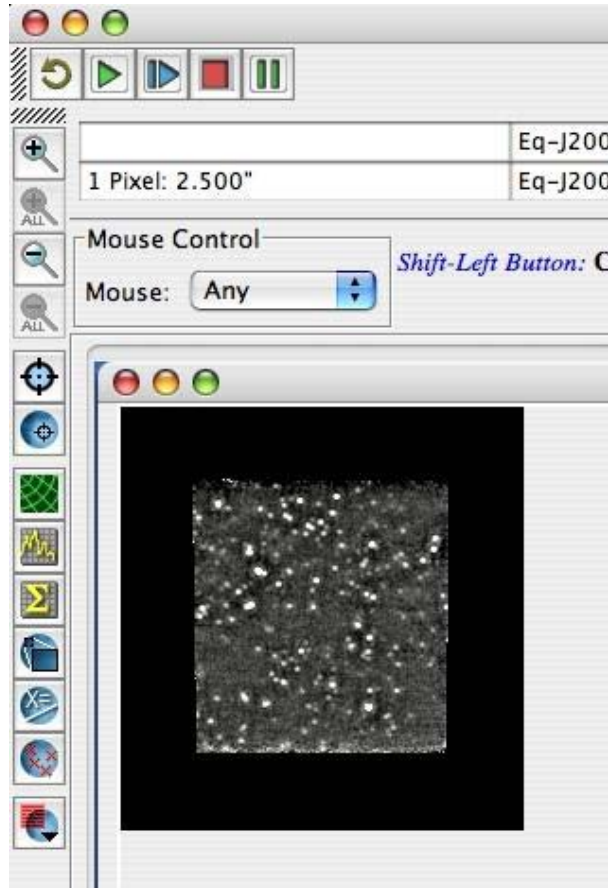
If we plot all of the sources in this table on top of the image, it looks like this:



The above image was made using ds9 regions files, but you can visualize the extracted catalog on top of the mosaic using MOPEX. To do so, click on Images-->Fits File Image. Then select the same image mosaic you used as input to APEX. In this case, the image is

r13810432/ch1/pbcd/Combine/mosaic.fits

The image will appear in the MOPEX window, but it will be quite small. The upper left-hand corner of the MOPEX window will look like this:



You can click on the top button on the left-hand side to zoom in.

Then, to overlay the catalog you just created, click on Overlays-->Catalog File. Then choose

r13810432/ch1/apex_output/mosaic_extract.tbl

Each detected source will be indicated with a red X by default, but a variety of symbols and colors are available.

Recipe 23. MOPEX: PRF Estimation and Point-Source Extraction

This recipe demonstrates how to select point sources for Point Response Function (PRF) estimation, run the PRF estimate tool, and then use the real PRF for point source extraction. Special care is taken in removing Airy rings of bright point sources, as they create false detections. The final products are a table of point source fluxes and a point source subtracted mosaic image. This recipe starts where Recipe 22 ends after making the science quality mosaic of a $z=0.7$ galaxy cluster using MOPEX.

23.1 Requirements

You will need to download MOPEX

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/tools/mopex>) in order to access the following scripts:

- *prf_estimate.pl*
- *apex_lframe.pl*
- *apex_qa.pl*

You will need to use these namelists:

- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/apex_lframe_step1.nl
- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/prf_estimate.nl
- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/apex_lframe_noring.nl
- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/apex_qa_noring.nl
- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/apex_lframe_step2a.nl
- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/apex_lframe_step2b.nl
- http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/apex_qa.nl

You will need the 24 microns data from AOR 13810432 as in the recipe in Recipe 22.

23.2 Step-by-Step Guide

1. Extract sources from the mosaic image that will be used to estimate the PRF:

```
unix% apex_lframe.pl -n apex_lframe_step1.nl
```

Use the PSP image to detect (set `use_psp_to_detect = 1`) and set the detection threshold to a large value to only include high SNR point sources (`Detection_Threshold = 20`). Use the MIPS 24 microns PRF found in the `cal/` directory of the MOPEX package.

The table of extracted sources is `apex_1frame_step1/mosaic_extract.tbl`.

2. Select the brightest point sources with the best fit:

```
unix% cd apex_1frame_step1
unix% select col all from mosaic_extract.tbl for "deblend = N and
chi2/dof < 30 and flux > 5000" into mosaic_extract_clean.tbl with
header
```

The new table of extracted sources is `mosaic_extract_clean.tbl`.

3. Visualize which sources you will use for PRF estimate with your favorite image viewer.

If using skyview, you can do:

```
unix% cd ..
unix% skyview
> pa data/main/mosaic.fits
> table apex_1frame_step1/mosaic_extract_clean.tbl mark $RA $Dec
circle red 1
```

4. Run PRF estimate. Use the input point source list `mosaic_extract_clean.tbl`:

```
unix% prf_estimate.pl -n prf_estimate.nl
```

Set the size of the PRF postage stamp image (`PostStamp_Xsize = 35`, `PostStamp_Ysize = 35`) and the sampling factor (`PRF_ResampleX_Factor = 4`, `PRF_ResampleY_Factor = 4`). If the PRF fluctuates a lot at the edge of the postage stamp image, set a region around the edge to zero (for an edge measuring 5 original pixels, set `Zero_Edge = 5`).

5. Copy the new PRF image to the cal directory:

```
unix% cp prf_estimate/PRF.fits cal/PRF_estimate.fits
```

6. Remove the first (and brightest) Airy ring. The Airy rings of the bright point sources create false detections.

```
unix% apex_1frame.pl -n apex_1frame_noring.nl
```

Use the PSP image to detect (set `use_psp_to_detect = 1`) and set the detection threshold to a large value to only include bright sources (`Detection_Threshold = 20`). Use the new PRF image. Set the detection threshold type to 'combo' (`Threshold_Type = 'combo'`).

7. Create a ring-subtracted image:

```
unix% apex_qa.pl -n apex_qa_noring.nl
```

Do not subtract the circular region of the PSF within the first minimum (HoleRadius = 21). This means that only the first Airy ring and beyond will be subtracted from the image. Set the radius for PRF subtraction (Radius = 44).

8. Run detect on the ringless image to create a list of sources:

```
unix% apex_lframe.pl -n apex_lframe_step2a.nl
```

Run the modules `run_medfilter`, `run_gaussnoise`, `run_pointsourceprob`, and `run_detect` (set them all = 1). Do not run the `run_sourcestimate` module at this point (set `run_sourcestimate` = 0). Use the PSP image to detect (set `use_psp_to_detect` = 1) and set the detection threshold to a low value (`Detection_Threshold` = 5).

9. Copy files to correct names for flux estimation (next step).

```
unix% cd apex_lframe_step2
unix% cp residual_mosaic_detect.tbl mosaic_detect.tbl
unix% cp residual_mosaic_minus_median.fits mosaic_minus_median.fits
unix% cp residual_mosaic_noise.fits mosaic_noise.fits
unix% cp residual_mosaic_PSP.fits mosaic_PSP.fits
```

Because we detected sources on the ringless image (*residual_mosaic.fits*), the output file names from step 7 have the root *residual_mosaic_*. However, we wish to measure the source fluxes on the original image (*mosaic.fits*). If we use an input mosaic called *mosaic.fits*, the extraction software will look for detection lists like *mosaic_detect.tbl*. Since we want the extraction software to use files like *residual_mosaic_detect.tbl*, we had to change some file names to trick APEX.

10. Run flux estimation using the original image:

```
unix% cd ..
unix% apex_lframe.pl -n apex_lframe_step2b.nl
```

Use the new PRF image for flux estimation. Select the fitting area (`Fitting_Area_X` = 9, `Fitting_Area_Y` = 9) and the active deblending parameters (`Max_Number_PS` = 3, `Chi_Threshold` = 10). Set the normalization radius for aperture correction (`Normalization_Radius` = 44).

11. Produce point source subtracted image to gauge how well the detection/extraction went:

```
unix% apex_qa.pl -n apex_qa.nl
```

12. Apply aperture correction

Recipe 24. MOPEX: MIPS 24 micron data from MIPS GAL

MIPSGAL (<http://mipsgal.ipac.caltech.edu>) is a survey of the inner 248 square degrees of the Galactic plane at 24 and 70 microns using the MIPS instrument aboard the Spitzer Space Telescope. The survey covers Galactic latitudes of $-1 < b < +1$ for Galactic longitudes of $l < 62$ and $l > 298$. In this recipe, we go through the steps required to extract photometry from a single MIPS GAL mosaic of size 1 degree on a side. We start with a mosaic that has been cleaned of artifacts such as jailbars (see the MIPS Instrument Handbook). We then outline the steps necessary to perform aperture photometry on the MIPS 24 microns MIPS GAL data, which is characterized by a complicated Galactic background.

24.1 Requirements

You must have MOPEX installed

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/tools/mopex/>).

You will need approximately 900Mb to go through this entire example, keeping all intermediate data products. To follow along with this demo, you should download this gzipped tar file:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/mipsgal24_cookbook_data.tar.gz The tarfile will unpack into a new directory called *mipsgal24_cookbook_data/*. Unless otherwise stated, you should issue all command-line arguments from this directory.

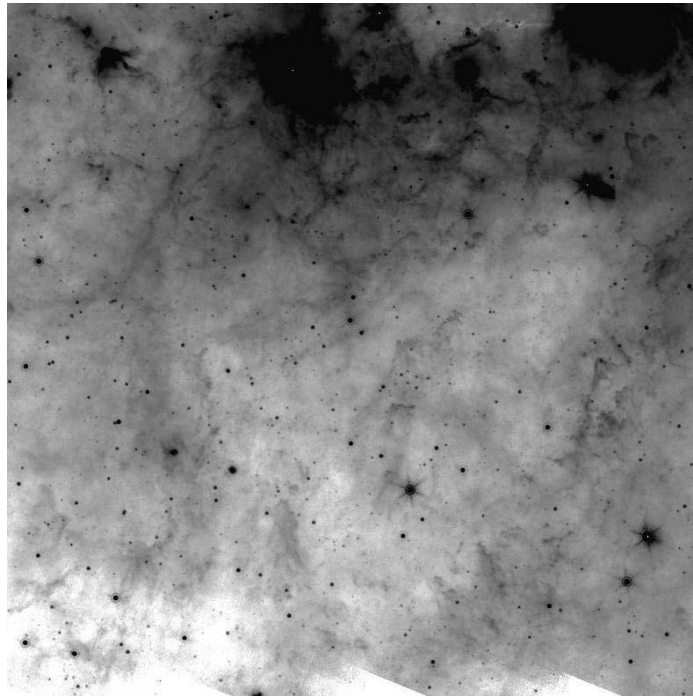
This tarfile includes:

1. *mosaic.fits* - The artifact-corrected MIPS GAL mosaic you will start with.
2. *cal/prf.fits* - The PRF that goes with *mosaic.fits*.
3. *cal/prf_bgsub.fits* - The filtered (background-subtracted) PRF
4. *cdf/apex_bri.nl* - The APEX namelist to detect and extract photometry for bright sources in the MIPS GAL mosaic
5. *cdf/apex_qa_ringless.nl* - The APEX_QA namelist to subtract 1st and 2nd Airy rings from around bright point sources found in the MIPS GAL mosaic.
6. *cdf/fit_radius.nl* - The APEX namelist to determine the radius around each source over which to fit the PRF during point-source detection.
7. *cdf/apex_det_fai.nl* - The APEX namelist to detect faint sources in the Airy-ring-subtracted MIPS GAL mosaic.
8. *cdf/apex_ext.nl* - The APEX namelist to extract photometry for sources that were previously detected in the MIPS GAL mosaic.

24.2 Examine the MIPS GAL mosaic.

Below is an example of a typical MIPS GAL field, after it has been corrected for artifacts such as jailbars. A FITS version (*mosaic.fits*) of this field is provided to you in the data tarball for this recipe.

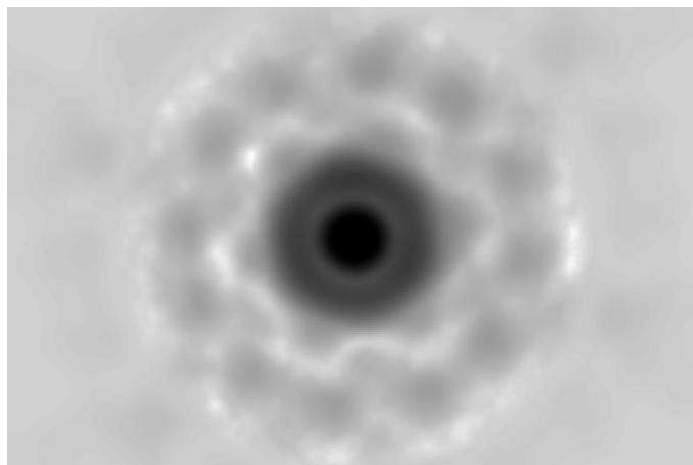
MIPSGAL surveys low galactic latitude fields characterized by high source densities. The strong interstellar dust emission along these lines of sight leads to spatially varying backgrounds, as seen below. These complicated backgrounds represent the primary challenge for extracting accurate photometry in these fields.



24.3 Derive the PRF.

In order to detect sources in a field with a complicated background, accurate knowledge of the Point Response Function (PRF; <http://irsa.ipac.caltech.edu/data/SPITZER/docs/mips/calibrationfiles/>) is critical. The better you know the shape of the sources you are interested in, the easier it is to differentiate them from the background. However, the presence of a complicated background can make measuring an accurate PRF from the data itself difficult. In particular, a spatially varying background can make the wings of the PRF difficult to estimate.

One solution is to measure the bright cores from the data, and join these to a theoretical shape for the wings. Using this method, we have created a PRF that is suitable for the MIPS GAL mosaic above. It is provided in the data tarball for this recipe. It is called *cal/prf.fits*. With a log stretch, it looks like this:



Note that the PRF was created from the above mosaic, but resampled by a factor of 4 in x and y. As a result, the PRF has a pixel scale of 0.3125 arcsec/pixel, while the mosaic above has a pixel scale of 1.25 arcsec/pixel.

24.4 Subtract the background from the PRF.

Because the background is very complicated, it can be difficult to subtract. The main issue is the size scale on which the background should be subtracted. A constant background is clearly not indicated by visual inspection of the MIPS GAL mosaic, since the background clearly contains a fair amount of structure. However, fitting a background over very small scales (on the order of the size of the sources), can result in subtracting some of the source flux.

After experimenting with the data, we found that the best results are achieved when the background is computed by taking a median filter of the image with an 11 pixel by 11 pixel filter, and this is reflected in all of the namelists included in this recipe. This is a rather small filter size, so we expect that some source flux will be subtracted. In a later step, we will add this source flux back in.

To properly extract sources from the background-subtracted mosaic, we need to use a background-subtracted PRF. We have provided the background-subtracted PRF in the data tarball for this recipe. It is called *cal/prf_bgsub.fits*.

If you would like to reproduce this result, keep in mind that the PRF is oversampled relative to the MIPS GAL mosaic.

24.5 Detect and extract bright sources in the background-subtracted mosaic.

We perform object detection and extraction in two steps: First we search for bright sources and remove them from the image. During a second pass, we search for fainter sources. The reason is that bright

sources are often surrounded by Airy rings. If not removed, these Airy rings are often detected as separate, faint objects.

In this step, we describe how to carry out the first step of this two-stage process. This is done with MOPEX and the following namelist:

```
cdf/apex_bri.nl
```

First, edit the above namelist to reflect the paths on your system.

Now, create an output directory: *bright_sources/*

After starting up the MOPEX GUI, you can load this namelist by clicking on File-->Read Name List. Then choose 'APEX single frame'. Finally, click on the green arrow on the top of the screen to run the namelist.

The primary outputs from this are:

bright_sources/mosaic_detect.tbl - list of detections

bright_sources/mosaic_extract.tbl - list of extractions

bright_sources/mosaic_minus_median_extract.fits - background-subtracted image upon which extraction was performed.

In MacOSX (Tiger), the above namelist may fail on select. To execute this module on the command line, type:

```
cd bright_sources
```

```
ssh
```

```
source /Applications/mopex/mopex-script-env.csh (substitute the path to your MOPEX installation directory)
```

```
select with complete_header from mosaic_extract_raw.tbl into  
mosaic_extract.tbl col  
"srcid,detid,RA,Dec,x,y,flux,delta_flux,SNR,chi2/dof,aperture1,  
aperture2,aperture3,aperture4,FitX,FitY"
```

```
cd ..
```

Both the detection and extraction tables may contain multiple detections of what is just a single true source. This can result if you set `Detection_Threshold` too low. In that case, the Airy rings around bright point sources will be detected as faint objects. Duplicates can also result from the shredding of soft saturated sources.

You will have to decide, based on your science goals, which sources qualify as duplications. MOPEX will not do this, so you will have to remove any potential duplicates from the above lists by hand, or with a simple piece of external code.

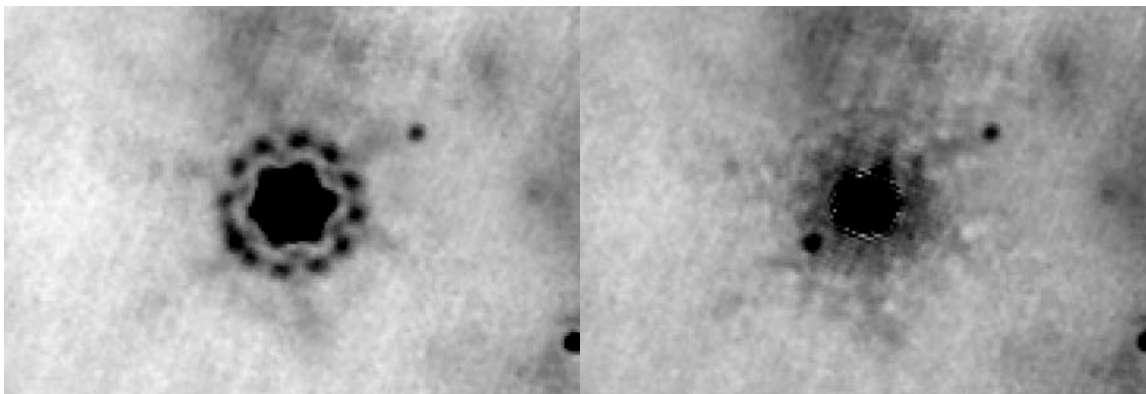
24.6 Remove Airy rings around bright sources in the background-subtracted mosaic.

Here we describe how to remove the Airy rings from around bright sources. This will prevent these rings from being detected as faint sources in the next step.

In the MOPEX GUI, simply read in the namelist *apex_qa_ringless.nl*, which is an APEX QA single-frame flow, and run as before.

The results are shown below. In the left image, we zoom in on a bright source in the MIPS GAL mosaic (*mosaic.fits*.) The Airy ring is prominent.

On the right, we show the same source with the Airy ring subtracted (*apex_qa_noring/Mosaic/residual_mosaic.fits*). The colorbars and scales are matched in both images. The Airy ring has been subtracted.



24.7 Detect faint sources in the background-subtracted, Airy-ring-subtracted mosaic.

Now that the Airy rings around bright sources have been removed, we can safely use MOPEX to detect faint sources.

To do this, we will use MOPEX and the following namelist:

apex_det_fai.nl

This namelist differs from the one you used to detect and extract bright sources (*apex_bri.nl*) in two ways:

1. We are only trying to detect sources in this step, whereas the previous namelist included modules to both detect AND extract sources.
2. The value of `Detection_Threshold` is much lower, since we are interested in finding faint sources now. Previously we were just looking for the brightest sources so we could subtract obvious Airy rings.

First edit *apex_det_fai.nl* so that the paths are appropriate for your system.

Now create an output directory called *detect_faint_sources/*. This is where the output from this run of MOPEX will go.

Now start the MOPEX GUI and load the namelist using the same procedure as before. MOPEX will ask you what type of flow this namelist contains. Click on APEX single frame.

The list of faint sources detected in this step can be found in *detect_faint_sources/residual_mosaic_detect.tbl*.

24.8 Extract photometry from the background-subtracted mosaic.

Now you should have a list of bright and faint sources in the image. First, you will have to concatenate these detection lists and remove any duplicates. Then you can use MOPEX to extract the resulting list of sources from the background-subtracted image. Unfortunately, the MOPEX GUI cannot be used to accomplish this task. Instead, we will use the command-line version of MOPEX. For the purposes of this example, we will just extract the faint detection list.

Make a directory called *extract_sources/*. This namelist includes only extraction modules. This is because we will use the detection lists we created in the last step.

To do this,

```
cp detect_faint_sources/residual_mosaic_detect.tbl
extract_sources/mosaic_detect.tbl
```

```
cp detect_faint_sources/residual_mosaic_minus_median_detect.fits
extract_sources/mosaic_minus_median_detect.fits
```

```
cp bright_sources/mosaic_minus_median_extract.fits
extract_sources/mosaic_minus_median_extract.fits
```

```

csh

source /Applications/mopex/mopex-script-env.csh
(substitute the path to your MOPEX installation directory)

cd extract_sources

select with complete_header from mosaic_detect.tbl into
mosaic_detect_sel.tbl for "blend_size < 10" col
"srcid,x,y,flux,blendid,blend_size"

cp mosaic_detect_sel.tbl mosaic_detect.tbl

cd ..

apex_1frame.pl -n fit_radius.nl

```

This step appends two columns to *extract_sources/mosaic_detect.tbl*: FitX and FitY. However, the headers for these two columns are in the wrong position. Make sure that there are 15 spaces between each pipe (|) before running the next step, or you will encounter an error in source estimate.

```
apex_1frame.pl -n apex_ext.nl
```

The primary output of this is *extract_sources/mosaic_extract.tbl*.

24.9 Apply scaling to photometry to correct for aggressive background subtraction

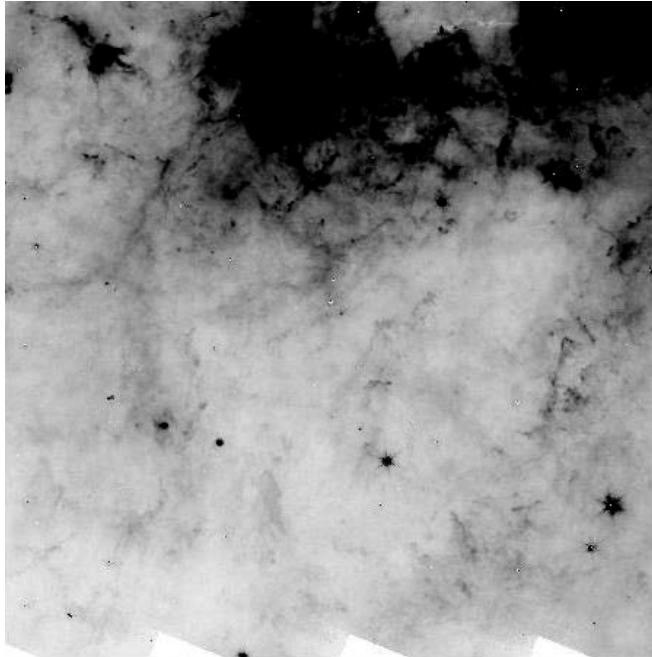
We now have flux densities computed using background-subtracted PRFs on our background-subtracted images. Because we used a relatively small background filtering scale, this photometry represents an underestimate of the real flux in each source. In order to account for this lost flux, we can calculate what fraction of the light within the PRF is lost during the filtering procedure.

MIPS routinely observes calibration stars. The 24 microns flux densities for these sources are well known, so they can be used to determine how much light is lost from a point source by using a background-subtracted PRF. The MIPS GAL team has run through the entire procedure above using these calibration numbers as input, and they have determined that the photometry needs to be scaled by a factor of 1.46.

For information on how to retrieve calibration observations to perform an analogous calculation for your own observations, please see the MIPS Instrument Handbook.

24.10 Quality Control

Finally, we must assess how well the above procedure has extracted the sources in our image. For this purpose, we can use *apex_qa.pl*. We used this module above in order to subtract Airy rings from around bright sources. It can also be used to subtract entire sources from our mosaic. We can then analyze the residuals to see if any of our detection, extraction, or duplicate removal procedures need to be revised. Below we show an example of the mosaic once all of the sources with good PRF fits have been subtracted.



Recipe 25. MOPEX: MIPS 70 micron mosaic of HII 173

In this recipe, we mosaic the FEPS Legacy Science program MIPS 70 microns small-field photometry observation of the Pleiades star HII 173. This recipe continues in a source extraction step.

25.1 Requirements

You will need to download MOPEX

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/mopex>), in order to run the script *mosaic.pl*. You will need to use this namelist:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/cookbook/files/mosaic_70um.nl.

Download the 70 microns BCD and post-BCD data for AOR 5315584. See Recipe 1 for instructions on downloading data via the Spitzer Archive interface.

25.2 Have a quick look at the data

The online (=automatically produced) 70 microns mosaics may not be good enough to start analyzing for science right away, but they are good enough to quickly assess your data - did you see something bright or not? For an initial look at your data, use ds9 or your favorite image viewer to examine the on-line mosaics that came with your data.

```
unix% cd /where/you/unpacked_data/r5315584/ch1/pbcd/  
unix% ds9 SPITZER_M1_5315584_0000_1_E429953_msaic.fits &
```

Note that, unlike for 24 micron observations, there are no "non-prime" data here. However, under pipeline version S11 onward, there are two mosaics - filtered and unfiltered. No source is apparent at the center, but off-center, there seems to be one. For completeness, we note that there are no 160 microns mosaics at all, because this array was never "prime," so all of these data are ancillary.

25.3 Set up lists of filenames and namelist files in preparation to remosaic data

Personally, I like to have just one copy of the *cal/* directory into which I can put the latest pmarks and the PRF (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/mips/calibrationfiles/>) I want to start from. Also copy the cdf namelist files for this example into a *cdf/* subdirectory.

```
unix% cd r5315584/ch1/bcd/  
unix% ln -s /where/you/unpacked_data/cal ./cal  
unix% mkdir cdf  
unix% mv /where/you/downloaded/mosaic_70um.nl ./cdf/
```

Lists that contain all of the BCDs, masks, and uncertainty files are needed as input to the *mosaic.pl* script.

```
unix% cd r5315584/ch2/bcd/  
unix% ls *_bcd.fits > InputImageList.txt  
unix% ls *bmask.fits > DmaskList.txt  
unix% ls *bunc.fits > SigmaList.txt  
unix% wc *.txt  
unix% nedit cdf/mosaic_70um.nl
```

Here you don't need to worry about serendipitous data or the first frame effect. You might want to check the length of each file (`wc *.txt`) to be sure that each list has the same number of files. If you find that that file has a lot more lines than the others, the following is probably what happened. Because the filtered and unfiltered BCDs have similar filename extensions, be sure that you are only putting the unfiltered (or just the filtered) into the *InputImageList.txt* file.

Edit the namelist to reflect where your cal files are located – be sure to use absolute pathnames!

25.4 Remosaic the data

```
unix% mosaic.pl -n mosaic_70um.nl  
OR  
unix% mkdir pbcd70  
unix% mosaic.pl -n mosaic_70um.nl > pbcd70/mosaic.log
```

Note that the namelist is *assumed* to be in the *cdf/* subdirectory, not in your current directory. If desired, redirect the output to a log file for your records. (actual mosaic log file:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/mosaic70.log>)

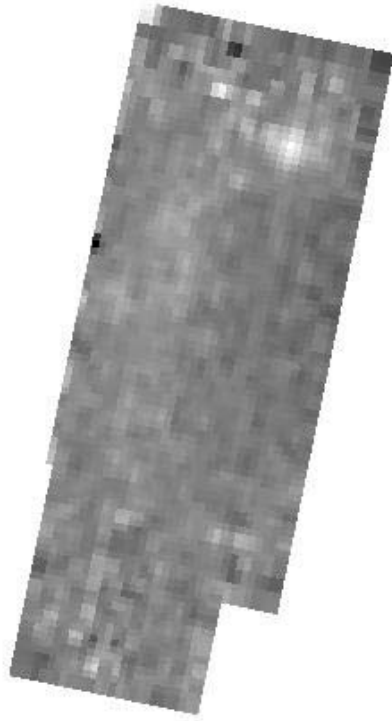
Common errors:

- "mosaic.pl: Command not found." means you have not sourced the csh when setting up the mopex package in your current shell (window).
- other "files not found" -- make sure you set all the paths correctly in the namelist, and that you're sitting in the correct directory when you call *mosaic.pl*.
- Error with "PSUtils" -- mopex_script_env.csh is not properly set up.

Check the mosaic.

```
unix% ds9 pbcd70/Combine/mosaic.fits &
```

Check the mosaic for anything that looks 'odd.' It should look similar to the unfiltered mosaic shown below (which came with your data, in the pbcd directory).



Recipe 26. MOPEX/APEX: MIPS 70um photometry of HII 173

In this recipe, we extract photometry for the FEPS Legacy Science program MIPS 70 microns observation in small-field photometry mode of the Pleiades star HII 173. This recipe continues from the mosaicking step (Recipe 25).

26.1 Requirements

You need to download MOPEX

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/mopex/>) in order to run the script *apex_lframe.pl*. You will need this namelist:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/cookbook/files/apex_lframe_MIPS70.nl

. And you will use the MOPEX output from Recipe 25.

26.2 Step-by-Step Guide

1. Get set up.

```
unix% cd r5315584/ch1/bcd/  
unix% mv /where/you/downloaded/apex_lframe_MIPS70.nl ./cdf/  
unix% nedit cdf/apex_lframe_MIPS70.nl
```

Edit the namelist to reflect the absolute pathnames, including the PRF

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/mips/calibrationfiles/>) you want to start from. Normally, it's best to create your own custom PRF for your own observations, but in this case, there aren't really any bright (known) sources, so we can't construct our own from this single observation. So, we have to depend on the version that came with the MOPEX package, or the version that is available online.

2. Extract sources!

```
unix% apex_lframe.pl -n apex_lframe_MIPS70.nl  
OR  
unix% mkdir pbcd70  
unix% apex_lframe.pl -n apex_lframe_MIPS70.nl > pbcd70/apex.log &
```

The star that is the target of this observation should be dead on in the center, but there doesn't appear to be anything there. It's very likely that this step will not catch anything at this location. (actual apex log file: <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/cookbook/files/apex70.log>) The parameters in the namelist are set such that the extraction step is included, and the final file with extracted sources is

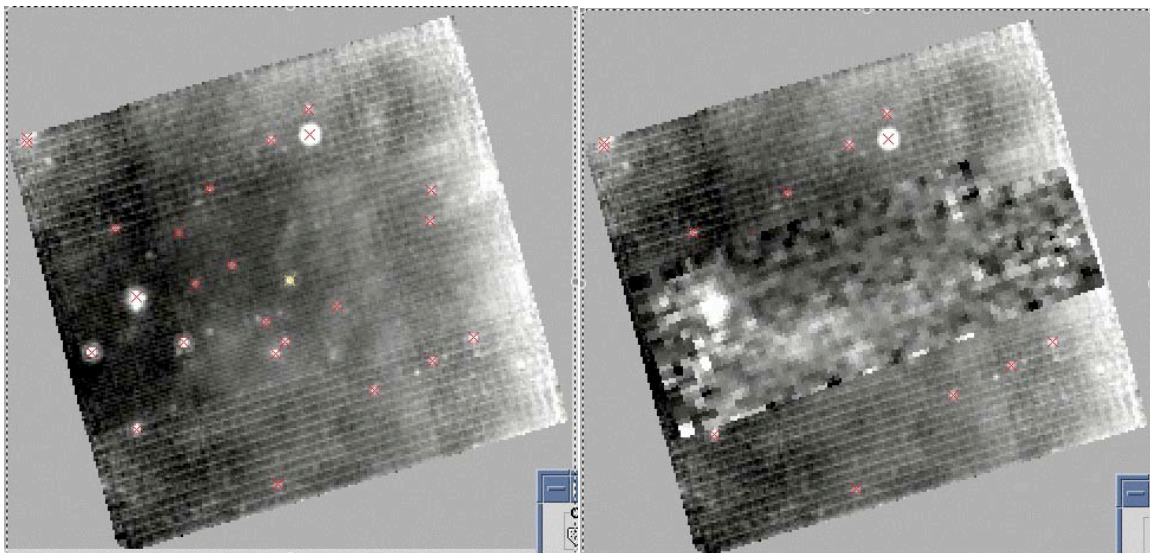
http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/mosaic70_extract.tbl. It completes successfully, but it indeed does not find any sources even meeting our selection criteria, much less at the center of the field.

3. Identify sources in list.

```
unix% spot
unix% nedit pbcd70/Combine/mosaic70_extract.tbl
```

In theory, had we found anything, we could have used the same trick as we did with 24 microns to find the object. However, in this case, there are no detections. We can still use Spot to create a nice fade between the two observations.

Start up Spot, and load in the 24 microns mosaic (images menu/FITS file option) -- no AORs needed. Be sure to load the file you just created -- *r5315584/ch1/bcd/pbcd24/Combine/mosaic.fits*. Next, load in the 70 microns mosaic you just created: *ch2/bcd/pbcd70/Combine/mosaic.fits*. Click on the little "%" icon on the right side of the spot window and change the transparency of the 70 microns image to watch the 24 microns image fade into and out of 70 microns. There sure doesn't seem to be anything at our source position, but there's something nice and bright near an edge at 70 microns, and it corresponds to a source seen at 24 microns -- it was source 197 in the 24 microns source list. There is very little redundancy at 70 microns in that position, so any photometry for this source will have very high errors.



Left image is 24 microns only; right image is showing the 70 micron image though the 24 micron (adjusting the opacity in Spot).

Recipe 27. MOPEX: MIPS 70 microns mosaic of the GFLS

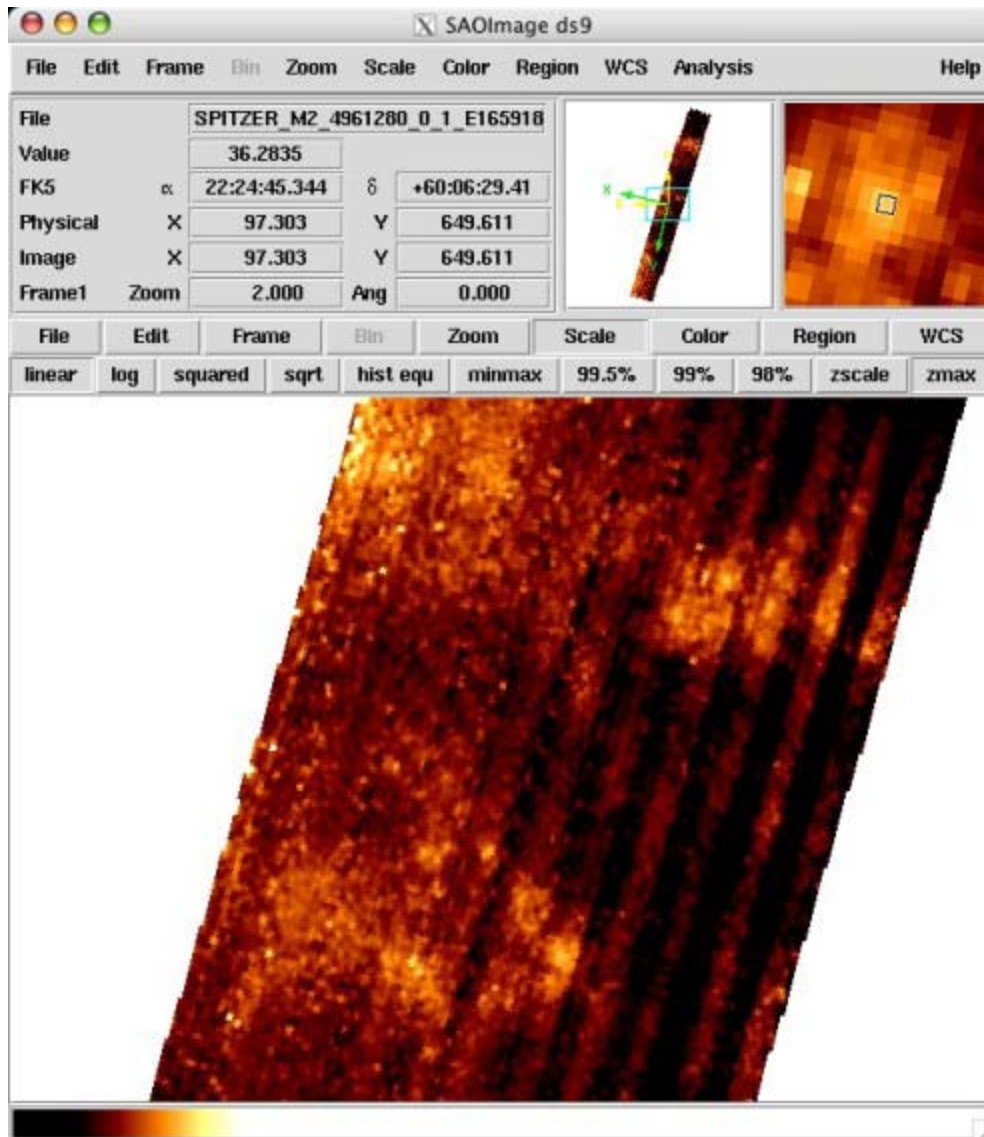
We will demonstrate use of the software tool MOPEX to create a mosaic of 70 microns images obtained with MIPS for the Galactic First Look Survey. This mosaic can be compared to the mosaic produced with MOPEX as part of the standard data processing pipeline.

27.1 Requirements

You will need access to the Spitzer archive tool and MOPEX (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/mopex/>). Download the 70 microns BCD and post-BCD data for the AOR 4961280. See Recipe 1 for instructions on using the Spitzer archive.

Have a quick look at the data:

If you wish, you can examine the mosaic produced automatically by the data processing pipeline. This is the file *r4961280/ch2/pbcd/SPITZER_M2_4961280_0_1_E165918_mosaic.fits*. You can use any FITS viewer to examine the mosaic image; an example using SAOIMAGE-DS9 is shown here (note that the mosaic covers a strip several degrees in length - only a small portion is seen in the magnified main viewing window):



27.2 Set up lists of filenames and namelist files in preparation to remosaic data

Obtain the appropriate namelist and pmask files to use with these data. The former is a parameter file for running MOPEX, while the latter is a mask of permanently bad pixels in the 70 microns array. These files are available on the SSC web site under MOPEX

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/tools/mopex/>).

Use your favorite text editor to ensure that the value of the PMASK_FILE_NAME parameter in the namelist is set to *mips70_pmask.fits*.

To create the mosaic with MOPEX, we will use the Basic Calibrated Data (BCD) files located in *r4961280/ch2/bcd/*. The first step is to create input lists for the BCD files (**_bcd.fits*), uncertainty images

(**_bunc.fits*), and mask files (**_bmask.fits*). The input order for the BCD files does not matter, but the *bunc* and *bmask* files must be listed in the same order. Sample input lists are available on the Spitzer web site:

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/bcdfiles>

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/buncfiles>

<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/cookbook/files/bmaskfiles>

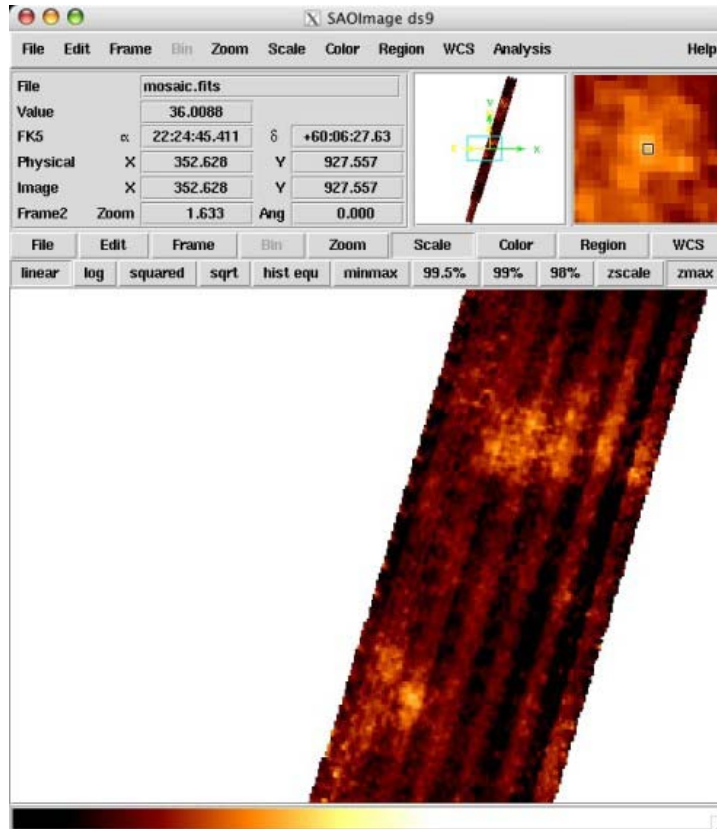
Store these files in the *r4961280/ch2/bcd/* directory (along with the *namelist* and *pmask* files). Note that in order to streamline the data processing for the purposes of this demonstration, we have only included half of the available images in the sample input lists. If you wish, you may repeat the demonstration using all of the available data.

27.3 Remosaic the data

We are now ready to run MOPEX. (Don't forget to source the MOPEX set-up file, *mopex.csh*.)

```
unix% mosaic.pl -I bcdfiles -S buncfiles -d bmaskfiles -O bcd_out -n  
mosaic_M2.nl
```

This will take several minutes to complete (e.g., about 15 minutes on a "standard" SunBlade 100). MOPEX creates a sub-directory called *bcd_out/*; the output image is located at *bcd_out/Combine/mosaic.fits*. Again, the mosaic image can be examined using your favorite FITS file viewer:



The "striping" of the mosaic image (which is visible in the output of both the automatic pipeline and MOPEX) is a data processing artifact caused by differences in detector response that are not removed by the pipeline at the BCD level. The striping in the GFLS data is particularly strong because the data were obtained prior to a bias change in the MIPS germanium arrays that was implemented specifically to address this artifact. This effect can be mitigated in the GFLS data by running MOPEX on the filtered BCD images (**_fbcdfits*) produced by the data processing pipeline, which are also located in the *r4961280/ch2/bcd/* directory.

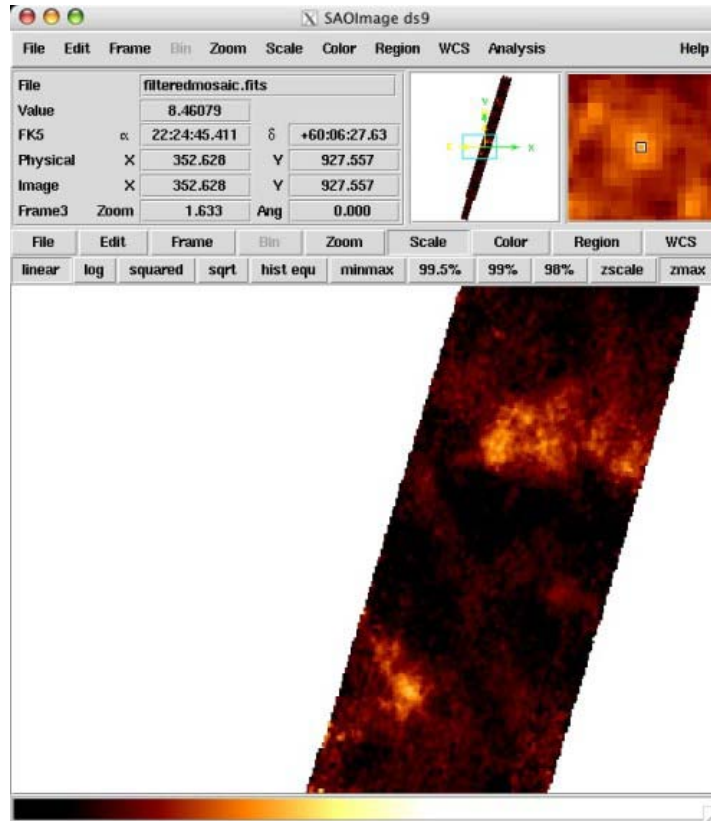
NOTE, however, that although the resultant mosaic will look better (see below), the filtering process renders the flux calibration of any extended emission in the image unreliable. Point source photometry should still be fine for targets fainter than 0.5 Jy at 70 microns (brighter targets will have some flux removed; i.e., they will actually be brighter than reported by photometry on the filtered mosaic image).

Construct a new BCD input list for the filtered files, *fbcdfiles*, (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/cookbook/files/fbcdfiles>), and run MOPEX again:

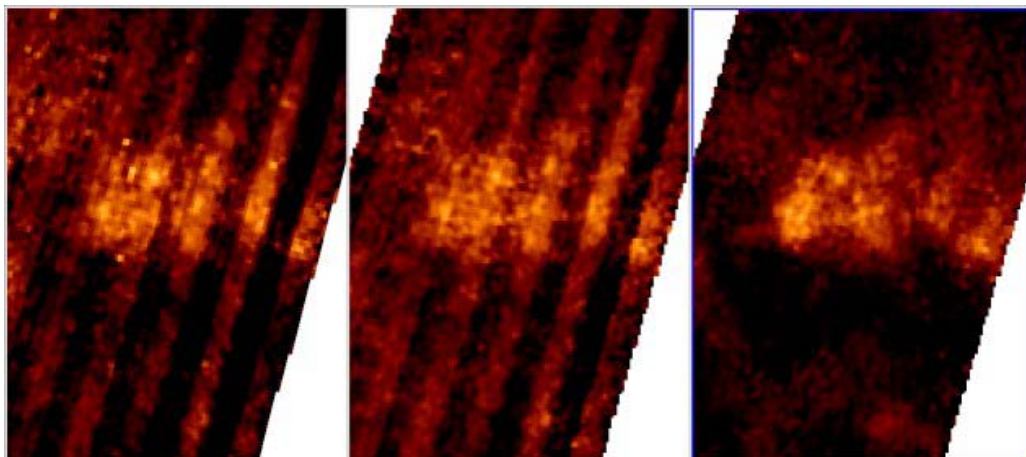
```
unix% mosaic.pl -I fbcdfiles -S buncfiles -d bmaskfiles -O fbcd_out -n mosaic_M2.nl
```

You can use the same input file lists for the *bmask* and *bunc* files, as well as the same namelist and *pmask* files, as were used for running MOPEX on the unfiltered BCD data. MOPEX creates a sub-directory

called *fbcd_out*; the output image is located at *fbcd_out/Combine/mosaic.fits*. Again, the mosaic image can be examined using your favorite FITS file viewer:



The following image shows a side-by-side comparison of the three mosaic images used in this demonstration: automatic data processing pipeline (left), MOPEX with unfiltered BCD data (middle), MOPEX with filtered BCD data (right). All three panels show the same region of sky at the same intensity scaling (keep in mind that only half of the available BCD data were processed with MOPEX in this demonstration, so the left panel shows slightly more sky - in the upper left corner - than the other two panels).



Recipe 28. MOPEX/APEX: Point-Source Photometry at 70um

This recipe illustrates how to measure photometry for point sources from MIPS 70 micron images with uniform background. We show how to use APEX, a photometry package provided by the Spitzer Science Center. APEX is a part of a general package called MOPEX: MOsaicker and Point source EXtractor. We STRONGLY recommend that users check the intermediate outputs, either images or tables, generated by APEX. This allows you to decide if the APEX input parameters are appropriate for your specific data set.

It is important to note that this tutorial is specifically written for MIPS 70 micron data, which has unresolved point sources and uniform background. The recommendations from this recipe are applicable to data sets similar to those from the Spitzer COSM^{ic} Origins Survey (SCOSMOS).

(<http://irsa.ipac.caltech.edu/data/SPITZER/docs/spitzermission/observingprograms/legacy/>) The SCOSMOS MIPS 70 micron data were taken in scan map mode with medium scan rate. The coverage per pixel is about 130-160. This type of 70 micron data were taken over a small area (0.25sq deg) in Spitzer Cycle 2, and were taken again over the full 2 sq. degrees in Spitzer Cycle 3. We use some of the Cycle-2 data for this recipe.

28.1 Requirements

You must have MOPEX (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisistools/tools/mopex/>) installed in order to follow along with this tutorial. MOPEX performs both image mosaicking and source photometry extraction. The source extraction part of MOPEX is called APEX. MOPEX/APEX can be used in either GUI mode or command line mode. For the first time users, we recommend the GUI version, which allows easy control and visualization of each processing step.

28.2 Download the example data

To follow the step-by-step instructions in this recipe, you will need to download the data set associated with Program 20070 (PI = D. Sanders). This program is Spitzer COSMOS (SCOSMOS) Legacy survey approved in Cycle 2. The MIPS 70 micron data were taken in scan mode. The final mosaic image was made by D. Frayer, a member of the SCOSMOS team, and a former member of the MIPS instrument support team. This recipe will use the data released by the SCOSMOS team via IPAC InfraRed Science Archive (IRSA). The data can be downloaded by going to <http://irsa.ipac.caltech.edu/data/COSMOS/images/spitzer/mips/>. From this link, you can download three images: *mips_70_go2_cov_10.fits* (coverage map), *mips_70_go2_sci_10.fits* (science image) and *mips_70_go2_unc_10.fits* (uncertainty image).

28.3 Extract Point Source Photometry

The primary purpose of this recipe is to present guidelines on how to extract point source photometry from a 70 micron mosaic with fairly uniform background. To begin, we recommend that users read the extensive documents on APEX software at <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/mopex/>. APEX can provide PSF-fitted photometry as well as aperture photometry (circular apertures). Because MIPS 70 micron data has a PSF of roughly 16 arcsec (FWHM), PSF fitting does a better job in separating blended sources, and also can handle noise more reliably for faint sources than using aperture photometry. For these reasons, we recommend obtaining PSF fitted photometry if your targeted science data are point sources with moderate to low signal-to-noise.

28.4 Introduction: several important issues

- ***Point source Response Function (PRF)***

When you download MOPEX, the MIPS 70 microns Point source Response Function (PRF) is a part of the package. This FITS file, *mips70_prf_mosaic_4.0_4x.fits*, is usually in a sub-directory called *cal/*. As its name indicates, this PRF was made for mosaicked 70 microns images. This PRF was made from a mosaic with a pixel scale of 4 arcseconds, generated from the Extragalactic First Look Survey (xFLS). It has a resampling factor of 4. It works well for sources of low and moderate brightnesses (<200 mJy). In addition, this PRF was derived to be large enough to go beyond the first Airy ring and its background was tuned to the zero level. If you use APEX for photometry extraction, you need to be sure that your 70 microns mosaic has the same pixel scale (4 arcsecond) as the PRF provided by APEX.

- ***Background***

The second important issue to understand before running APEX is the background level in your mosaic image. This point is briefly mentioned in the previous discussion of the PRF. The principle is that whatever PRF you use has to be consistent with the mosaic image on which source extraction will be carried out. In addition to matching pixel scales as mentioned above, the background in the PRF image must match that in the mosaic image. The delivered 70 microns PRF image included in the APEX package was made to have a background level of zero. Therefore, users should ensure that within APEX the PRF fitting procedure is performed on an image which has a zero-background level. To achieve this goal, you can either directly input a background-subtracted image into APEX; or you can input a mosaic with non-zero background, then perform the background subtraction before measuring PRF-fitted photometry.

Here are the two ways to make background subtracted images from which PRF-fitted photometry can be measured:

1. **Manually subtract the background before starting APEX :** One can derive the background level by finding the median of the data distribution for source-free pixels within the mosaic. If one uses all pixels within a region, one can first apply multiple iterations of sigma rejection of outlier pixels before computing the median value. One can also fit a Gaussian to the central core of the data distribution after clipping off the wings.
2. **Use the APEX module Extract MedFilter:** It is possible to use this module to produce a reasonable background subtracted image for PRF fitted photometry. The trick is to make sure that the box size is big enough so that you are not including positive signals from sources, thus over-subtracting the background. With the SCOSMOS 70 microns data, we found that we can tune the APEX median filter parameters to produce acceptable background subtracted images. The specific values for the parameters in this module are discussed below.

For SCOSMOS data, we have tested both of these methods, and found very small differences. It is important to note that we run **APEX Med Filter** several times to identify the optimal parameters by comparing the results from both methods. For any other data, we strongly recommend users to examine the output from **MedFilter** , and make comparisons of results from two methods. It is important to be sure that the image used for the PRF-fitted photometry has zero background.

- **Noise Map**

In general, you will encounter three noise maps after you have used MOPEX to put together all individual exposures into a single mosaic.

The noise map with an extension of **_unc.fits* is usually based on the SSC pipeline error propagation using individual *bunc.fits* images from the BCD products. The pipeline uncertainties were tuned to slightly under-estimate the true noise. This was tuned to be low to ensure that on-line pipeline processing can produce reasonable outlier rejection. We do not recommend to use this type of noise map for source detection and computing SNR.

The second noise image is measured by the standard deviation of the pixel stack at each sky position. This noise image is one of the outputs from the image mosaic tool MOPEX, and has an extension **std.fits*. When your data set has good coverage, this type of noise map is usually much closer to the true sky noise level.

The third noise estimate is from APEX module **GAUSSNOISE**. This method involves measuring the pixel value distribution within a sliding box. In this method, the user can clip out a certain number of pixels whose data values deviate significantly from the median value. We will demonstrate the specific parameter settings in Section II. This noise image represents spatial pixel-to-pixel noise.

APEX uses noise images for two purposes: one is to determine the detection threshold, another is to compute the signal-to-noise ratio for the PRF-fitted photometry. By default, APEX uses the noise estimated from the **GAUSSNOISE** module to determine sources above the specified detection threshold as well as for computing the SNR of the resulting photometry. The latest version of APEX also offers another option which allows users to choose a different noise image for SNR computation. This parameter

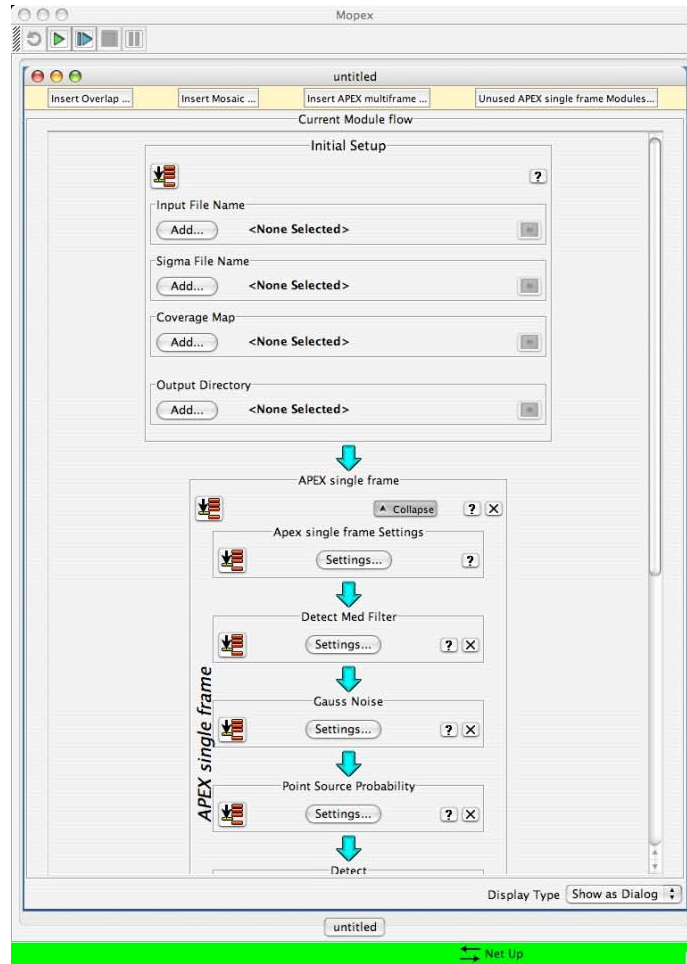
is in the APEX single frame setting --- a switch allows one to use `SIGMA_FILE` to estimate SNR. This `SIGMA_FILE` can be defined by users to be any preferred noise map.

For the COSMOS data, we found that the noise image made by `GAUSSNOISE` is a little bit higher than *mips_70_go2_unc_10.fits*. In principle, if module `GAUSSNOISE` is properly tuned, its result should be close to the noise image measured from pixel stack (method 2 above). We recommend that users make comparisons among these noise images and make the appropriate selection for their own data. For the specific example here, we used the noise map estimated from `GAUSSNOISE` for detection and computing the SNR during the source fitting.

28.5 How to Run APEX

With the data set covering the full COSMOS area downloaded from IRSA, you can cut out the central region with deep data, and put a prefix 'sub' to all the file names. We run APEX with the following basic steps:

- 1. Start a new APEX single-frame pipeline.** After starting up MOPEX, go to the main menu and select File-->New Apex Single-Frame Pipeline. In the window that pops up, select "APEX 1frame, MIPS 70 microns". The previously-empty MOPEX window will now be filled with a module flow consisting of "Initial Setup" and "APEX single frame". It will look like this:



2. Set parameters.

Only the parameters that require a change from the template default are listed below.

Initial Settings:

Input File Name	<i>mips70/sub_mips_70_go2_sci_10_0bck.fits</i> The background in this image has been manually subtracted and set to zero. See below for a detailed discussion.
Sigma File Name	<i>mips70/sub_mips_70_go2_unc_10.fits</i>
Coverage Map	<i>mips70/sub_mips_70_go2_cov_10.fits</i>
Output Directory	<i>mips70/apex_output/</i>

Although we specify *sub_mips_70_go2_unc_10.fits* as the input noise map, during the extraction and SNR calculation, we recomputed noise within APEX using **GAUSSNOISE** module. For details, see below.

APEX single frame settings:

PRF_File_Name /*Applications/mopex/cal/mips70_prf_mosaic_4.0_4x.fits*
You will have to modify your path to reflect your mopex installation directory

Use PSP to detect input image without background subtraction

Here you need to choose the type of image on which the source detection is performed. In the APEX single frame setting, there are 4 options:

- (1) input image w/o background subtraction, which means source detection can directly run on it without any additional background subtraction.
- (2) Filtered image, which is essentially smoothed by a gaussian similar to the MIPS 70 micron PSF.
- (3) PSP image, which is Point Source Probability image, generated by Point Source probability module.
- (4) background subtracted input image, which will use module **Detect Med Filter** to produce a zero-background image.

For the detailed explanation for each of these options, please refer to <http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysistools/tools/mopex/>.

For our specific data set, we choose option (1), which allows the **Detect** module to directly run detection on it. With this option, you can directly fit the PRF to the image to obtain point source photometry. If you choose option (4), then APEX will need to subtract the background image. We have tested both options, and found that our manually background subtracted image produces slightly better (<2%) fluxes for bright sources than option (4), which uses **Detect Medfilter** to subtract the background.

For the shallow xFLS data, option (3), the filtered image, gives good results in detection, particularly for not producing false detections near bright sources. Option (4), PSP image, smooths data too much. A detailed discussion of the shallow data can also be found for the publicly released 70 micron image and catalogs for the xFLS data. Please refer to Frayer, D. et al. 2005, AJ, 131, 249. For the COSMOS data, where source blending is an issue, we use the input image without background subtraction since the filtered image can smooth out faint sources near a bright source. The down-side of this choice is that some false detections near most bright sources can not be avoided. These false detections can be cleaned up manually.

use background subtracted image for fitting = no

use data unc for fitted SNR = no

This option is set to no by default when you use **GAUSSNOISE** to estimate a better noise map for fitted SNR calculation. However, if you have a better noise map defined by **SIGMA_FILE**, you can set this option to yes for calculating SNR in the PRF fitting procedure.

Detect Med Filter Settings:

Window X (Y) 100 (100)

Outliers/Window 500

If the input image to the **DETECT** module is option (4), background subtracted image, then you need to run module **Detect MedFilter** to set the background level to zero before passing it to **DETECT**. Here is the explanation on how to set the parameters, and what we have tested for the SCOSMOS data set. This module produces a reasonable background subtracted image if the box size is set correctly. For MIPS 70 micron mosaic images featuring a pixel size of 4 arcsecond per pixel, one should use a box of 100 pixel by 100 pixel with the number of outliers set to 500. This rejects about 5% of bright pixels. This parameter is tuned for the moderate to deep MIPS 70 microns images taken by the SCOSMOS survey. The total integration time is about 1500 seconds. Boxes smaller than this will produce a background image with bright patches in regions with clusters of sources. This causes over-subtraction of the background for bright sources. This will affect the flux densities of bright sources at a level of a few percent. Obviously for shallower data with a lower surface number density of sources, the median box size could be smaller than what was used for the COSMOS data. Users need to experiment with their data to find the right parameters.

GAUSS NOISE Settings:

Window X (Y) 100 (100)

Outliers/Window 500

This module estimates the noise from a distribution of pixel values measured from a sliding box with a user-defined number of outliers. This noise image represents spatial pixel-to-pixel noise. By default, APEX uses the noise estimated from the GAUSSNOISE module to determine sources above the specified detection threshold as well as for computing the SNR of the resulting photometry.

DETECT

Detection_Max_Area 25

This is a number of pixels within the FWHM of the beam. Here we assume that the pixel size is 4 arcseconds. It is recommended that you set this value small to avoid breaking up individual sources.

Detection_Min_Area 6

We suggest that this parameter is not set to 1 to avoid spurious single pixel sources.

Extended_Object_Area 10000

You should set this parameter very large to avoid missing bright nearby sources by mistaking classifying the entire cluster region as an extended object.

Detection_Threshold	2.5	This threshold depends on the input image type and use_psp parameter. For the filtered and PSP images, one needs a larger threshold parameter than used for the raw image to reach similar source SNRs. We do not recommend digging too deep with APEX; we recommend only going deep enough to reach the desired depth in source SNRs such that number of sources in the raw extraction table is about 1.5--2 times that in the selected extraction table. If you set the threshold too low (APEX will take a long time) and can assign real source flux to nearby spurious noise peaks. If you set the threshold too high, then you can be incomplete at your selected SNR cutoff (e.g., when the raw table has a similar number of sources as your selected table).
Input_Type	"image_input"	We use Threshold_Type = 'peak'. This requires a pixel to be larger than the 8 adjacent pixels before declaring a detection (for Peaks_Radius=1). For close blends, the peaks of the fainter source(s) in the blend(s) may not be found. Also, APEX factors in the coverage map in determining whether a pixel is a peak (in rare cases the peak image pixel is not identified due to local variation of coverage).
Minimum Coverage	4	

EXTRACTMEDFILTER

The parameters in this module are set with the same values as in **Detect Med Filter**

SOURCESTIMATE

Fitting_area_x/y	5	The units are pixels. This is set to be similar to the FWHM, so that APEX performs the fitting over a significant fraction of the area around the peak. If this parameter is set too low, there will not be enough pixels and APEX may derive a position off the real peak.
Max_Number_PS	1	This is set to 1 so that APEX fits only one source to each peak.

APERTURE PHOTOMETRY:

Number of Apertures	3	Number of circular apertures to draw around each source, between 1 and 5.
Aperture Radius 1, 2, 3	8.0, 10, 12.0	Radius of aperture N in pixels.
Inner and Outer Radius	15, 30	

28.6 APEX parameters for COSMOS Data

Here we provide the exact parameters that were used for the COSMOS MIPS 70 microns mosaicked image. This set of parameters has produced a reasonable photometry catalog.

```

compute_uncertainties_internally = 0
have_uncertainties = 1
run_detect_medfilter = 0
run_gaussnoise = 1
run_pointsourceprob = 0
run_bright_detect = 0
run_detect = 1
run_select_detect = 0
run_extract_medfilter = 0
run_fit_radius = 0
run_sourcestimate = 1
run_aperture = 1
run_select = 1

OUTPUT_DIR = mips70/apex_output
INPUT_FILE_NAME = mips70/sub_mips_70_go2_sci_10_0bck.fits
SIGMA_FILE_NAME = mips70/sub_mips_70_go2_unc_10.fits
COVERAGE_MAP = mips70/sub_mips_70_go2_cov_10.fits
PRF_file_name = /Applications/mopex/cal/mips70_prf_mosaic_4.0_4x.fits

use_refined_pointing = 0
use_data_unc_for_fitted_SNR = 0
use_background_subtracted_image_for_fitting = 0
PMask_Fatal_BitPattern = 0
use_background_subtracted_image_for_aperture = 0

use_psp_to_detect = -1
use_extract_table_for_aperture = 1
RMask_Fatal_BitPattern = 0

```

```

DCE_Status_Mask_Fatal_BitPattern = 0

select_conditions = "SNR > 4.0 and deblend ! NO and deblend ! PO and
deblend !
AO and deblend ! PAO"
select_columns =
"srcid,detid,RA,delta_RA,Dec,delta_Dec,delta_RAD,x,delta_x,y,delta_y,
delta_xy,fl
ux,delta_flux,chi2/dof,ps_chi2/dof,status,SNR,deblend,aperture1,
aperture2,apertu
re3,bad_pix1,bad_pix2,bad_pix3,ap_unc1,ap_unc2,ap_unc3"

use_bright_object_mask = 0
PROBABILITY_THRESHOLD = 0.0

&SNESTIMATORIN (THIS MODULE IS TURNED OFF)
&END

&DETECT_MEDFILTER
Window_Y = 100,
N_Outliers_Per_Window = 500,
Window_X = 100,
Max_Bad_Pixels_OutputImage = 1.0,
Min_GoodNeighbors_Number = 4,
Min_Good_Pixels_In_Window = 9,
&END

&GAUSSNOISE
Window_Y = 100,
Max_BadPixels_OutputImage = 1.0,
N_Outliers_Per_Window = 500,
Window_X = 100,
Min_GoodNeighbors_Number = 4,
Min_Good_Pixels_In_Window = 9,
&END

&POINTSOURCEPROB
PRF_ResampleY_Factor = 4,
Apriori_Probability = 0.1,
PRF_Xsize = 11,
PRF_Ysize = 11,
PRF_ResampleX_Factor = 4,
&END

```

```
&BRIGHT_DETECT (THIS MODULE IS TURNED OFF)
&END
```

```
&DETECT
Detection_Threshold = 2.5,
Input_Type = 'image_input',
Detection_Max_Area = 25,
Min_Coverage = 4.0,
Detection_Min_Area = 6,
Threshold_Type = 'peak',
Extended_Object_Area = 10000,
&END
```

```
&EXTRACT_MEDFILTER
(This module is turned off for this setting. If the initial input
image has none zero background, this module needs to be on)
&END
```

```
&FIT_RADIUS
&END
```

```
&SOURCESTIMATE
MinimizeFtolSuccess = 1.0E-4,
PRF_ResampleX_Factor = 4,
Max_Number_PS = 1,
Chi_Threshold = 3.0,
Background_Fit = 0,
Max_N_Iteration = 50000,
Fitting_Area_Y = 5,
InputType = 'image_list',
PRF_ResampleY_Factor = 4,
DitherPixelFraction = 0.1,
Max_N_Success_Iteration = 1000,
Chi2_Improvement = 1.0,
DitherFluxFraction = 0.8,
MinimizeFtol = 1.0E-4,
Fitting_Area_X = 5,
DeblendDitherPixelFraction = 1.0,
N_Edge = 4,
Random_Fit = 0,
&END
```

```
&APERTURE
Annulus_Compute_Type = 'median',
N_Apertures = 3,
```

```
Use_Annulus = 1,  
Aperture_Radius_1 = 8.0,  
Aperture_Radius_2 = 10.0,  
Aperture_Radius_3 = 12.0,  
Min_Number_Pixels = 10,  
Inner_Radius = 15.0,  
Outer_Radius = 30.0,  
&END
```

```
&SELECT  
&END
```

```
#END
```

Recipe 29. GeRT: Bad Stim Flash Correction in MIPS-70 data

In this recipe, we use the GeRT to correct for a poor stimflash calibration in 70 microns photometry observations, before running MOPEX to create the final mosaic. For those users who are not interested in diffuse extended emission, such as the ISM, this recipe then continues in a filtering step.

29.1 Requirements

You will need access to the Spitzer archive tool to download the example data set. You will also need to download MOPEX (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/mopex>), in order to mosaic the individual data frames.

You will need to download The GeRT (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/gert>) in order to correct for bad stim flashes in your data.

The data should be retrieved from the Spitzer archive (see Recipe 1). Search the archive by AOR and download the post-BCD, BCD and raw data for the following dataset: AOR 5568512.

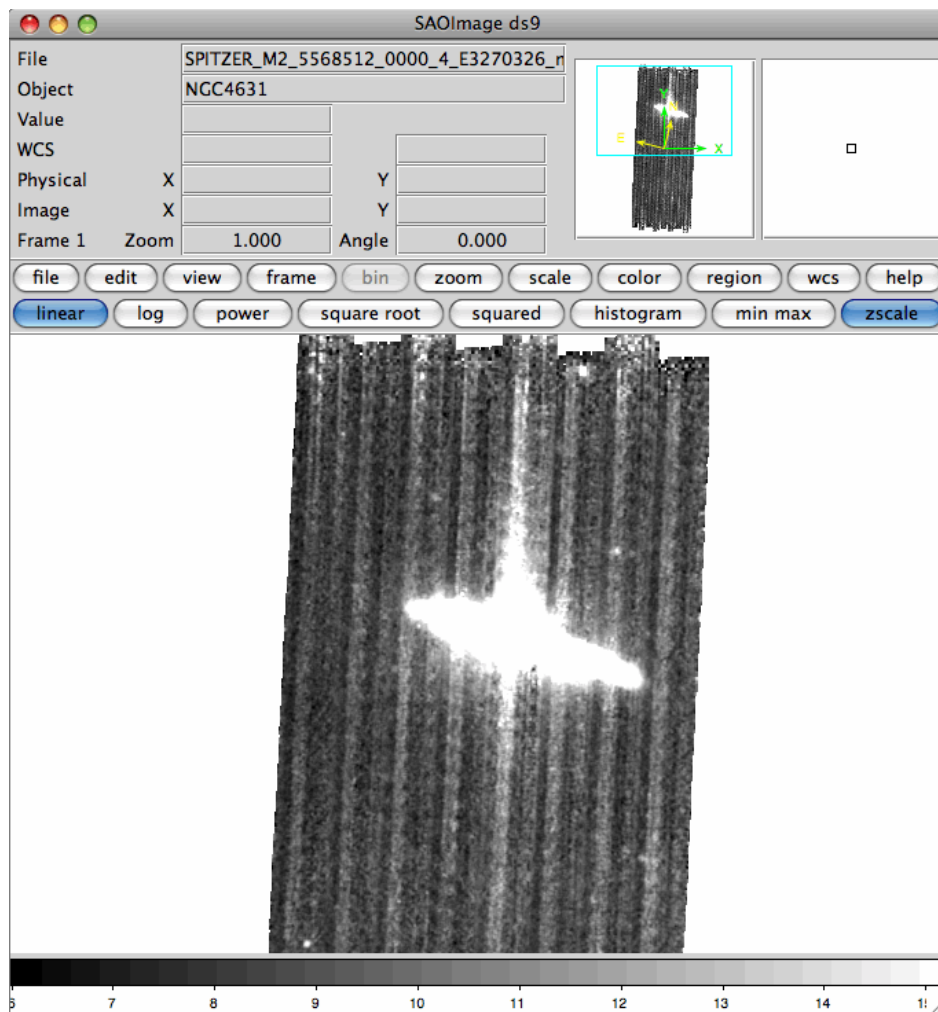
You will need approximately 160 MB of space available to download and unpack the data. Once you have downloaded the files to your disk, unzip them into a new working directory. The data will be unzipped into subdirectories *r5568512/ch2/bcd/*, *r5568512/ch2/pbcd/* and *r5568512/ch2/raw/*.

29.2 Have a quick look at the data

The online (=automatically produced) 70 micron mosaics may not be good enough to start analyzing for science right away, but they are good enough to quickly assess your data - did you see something bright or not? For an initial look at your data, use ds9 or your favorite image viewer to examine the on-line mosaics that came with your data.

```
unix% cd /where/you/unpacked_data/r5568512/ch2/pbcd/  
unix% ds9 SPITZER_M2_5568512_0000_4_E3270326_maic.fits &
```

The view in SAOIMAGE-DS9 is below. There are two immediately noticeable problems with this mosaic. Firstly, the vertical striping across the detector. This can be mitigated with careful re-filtering of the BCDs, and is explored in the next recipe, GeRT: Filter Extended Sources in MIPS-70 data. The second issue is the "smearing" of the galaxy up the column on which it appears. This is due to a poorly calibrated stim response, caused by the stim flash being taken while the field of view was on top of the bright galaxy. This cookbook addresses how to correct for this type of poor stim calibration by re-creating the BCDs from the raw files using the GeRT.



You can download this as a FITS image here :

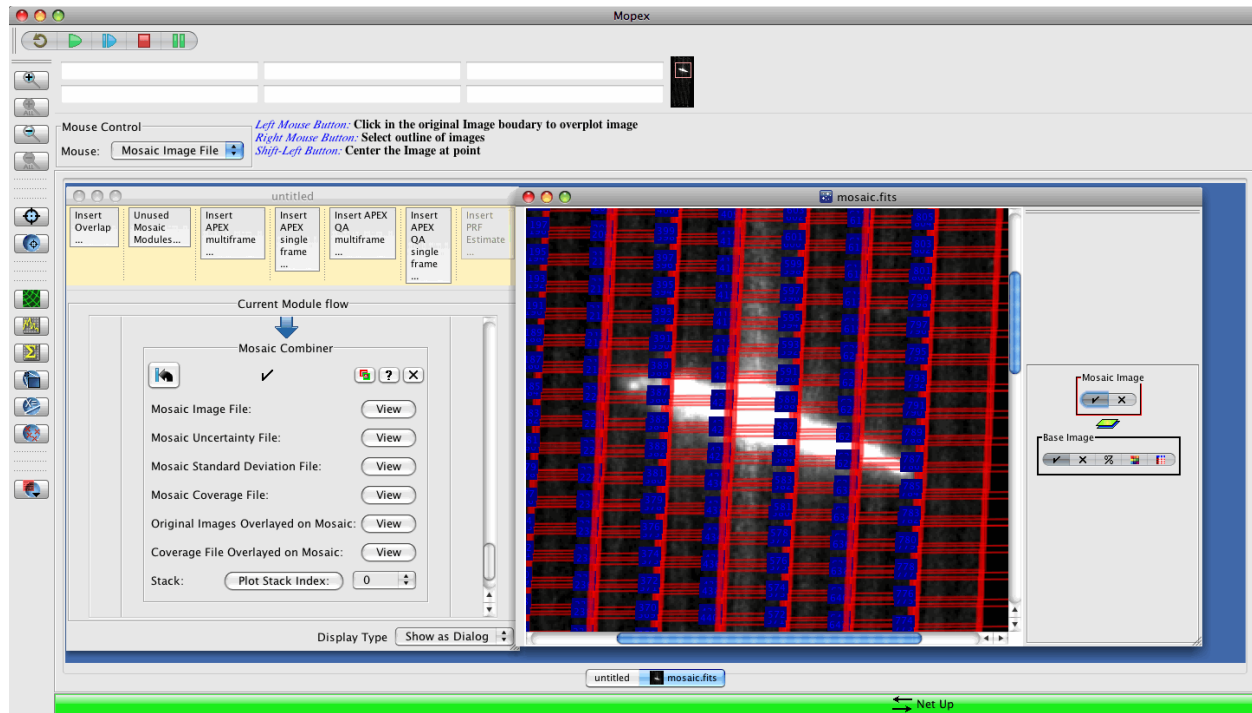
http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/mips70_stimcorr_pbcd.fits.

29.3 Mosaic the BCDs to identify the bad stim flash frames

Start the MOPEX GUI and load the template mosaic flow for MIPS 70 microns data (go to File > New Mosaic Pipeline > Mosaic, MIPS 70 microns and hit "ok"). Set up the flow by setting the Image Stack File to point to the file listing the **bcd.fits* files and choosing an Output Directory. Be patient when trying to load the Image Stack File - the directory contains a lot of images, and it takes a while for MOPEX to load the directory contents into the file selection window. Next you need to click on "Optional Input & Mask Files" and set the Sigma List File and the DCE Status Mask File to point to the lists of the **bunc.fits* and **bmask.fits* files, respectively. Again, be patient, as it takes a while for MOPEX to display the directory contents. The Pmask FITS File should be set to the MIPS 70 microns PMask file found in the *cal/* directory of your MOPEX distribution (*cal/mips70_pmask.fits*), but the other input files can be

left blank. Click "OK" and then start the Mosaic flow by clicking on the green arrow in the top right corner of the MOPEX window.

Once the mosaicking process has finished, click on "View" next to "Mosaic Image File". This will bring up the combined mosaic image and ask you whether you'd like to see the BCD outlines. Click "Yes" to overlay the BCDs on the mosaicked image.



We are trying to identify the stim flashes that have been taken on top of the bright galaxy, so zoom in on that region of the mosaic using the zoom function on the left hand toolbar. Once you have zoomed in on the galaxy, you should click on the image somewhere over the galaxy. This will bring up a dialogue box listing all of the BCDs that contributed to the pixel that you clicked on. Compare this list of contributing BCDs to the list of stim flash files that you made above. If any of the contributing BCDs are stim flash files then you should make a note of them so you can remove them from the GeRT input lists in the next step. Then hit "Cancel" to avoid opening all of the individual BCDs. Repeat this process over the area covered by the galaxy until you have identified all the stim flash frames. You should have found 5 files:

*SPITZER_M2_5568512_0002_0025_4_bcd.fits, SPITZER_M2_5568512_0003_0075_4_bcd.fits,
SPITZER_M2_5568512_0004_0025_4_bcd.fits, SPITZER_M2_5568512_0005_0075_4_bcd.fits,
SPITZER_M2_5568512_0006_0025_4_bcd.fits.*

29.4 Set up the files for input into the GeRT

Once we have identified the bad stim frames in Section 25.3, we need to recreate the BCDs from the raw files, skipping over the poorly-calibrated stims. To do this we use the GeRT. Start by moving to the directory containing the raw files:

```
unix% cd /where/you/unpacked_data/r5568512/ch2/raw/
```

Now list all of the raw files into a single text file:

```
unix% ls *raw.fits > rawlist.txt
```

and edit it to remove the bad stim flash files using emacs or your favorite editor.

29.5 Running the GeRT

Make sure that you have set up and tested the GeRT before you try to run it, following the test procedure outlined in the GeRT Manual.

Set up the paths for the GeRT by sourcing the setup script. The directory name for the GeRT package depends on your system, but for the Mac Intel you would type e.g.:

```
unix% source ~/Software/GeRT_Mac_070904/gert_setup.csh
```

Run the GeRT on your raw files to create the new BCDs, giving the input list of raw files and the desired output directory as arguments on the command line:

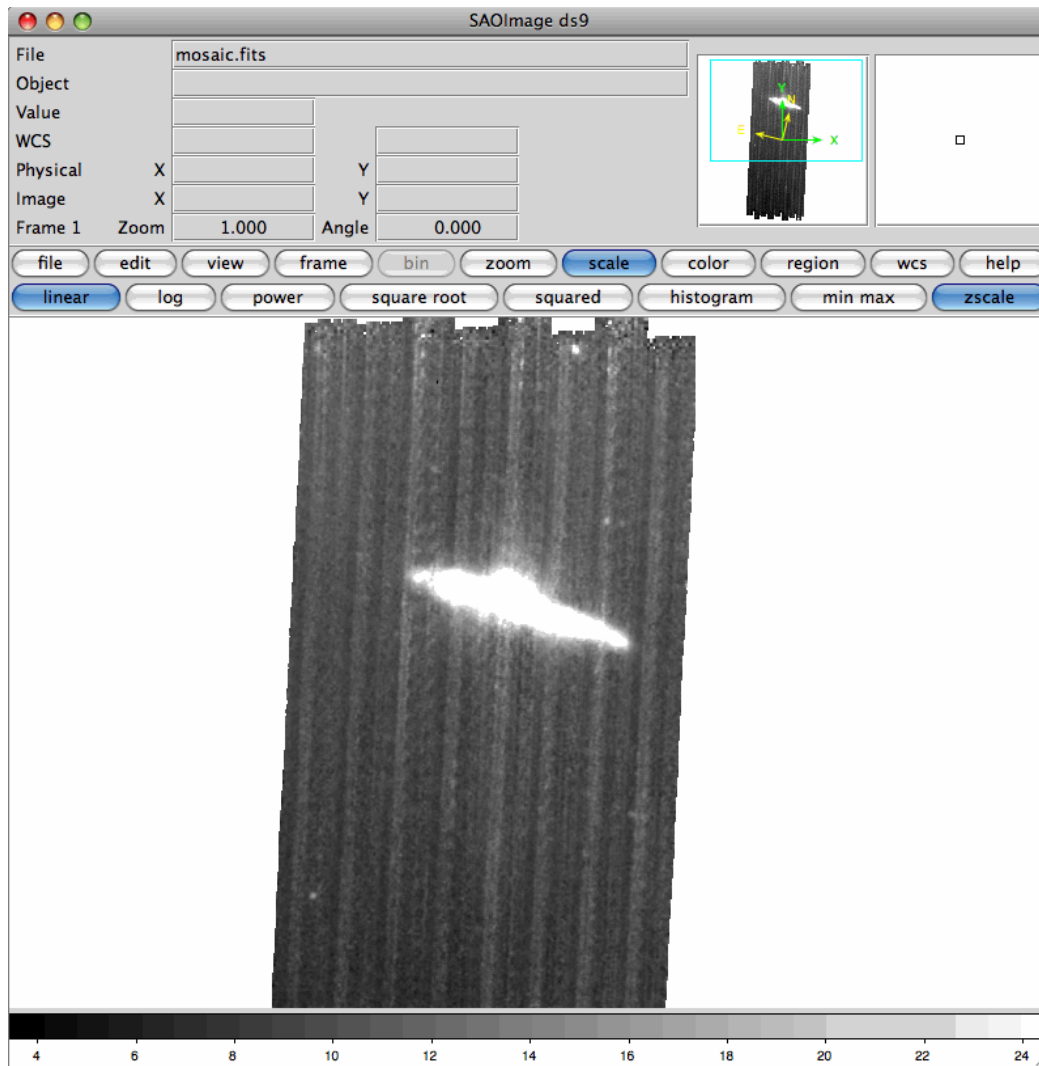
```
unix% $WRAPDIR/gert.pl rawfiles.lis OUTdir
```

The GeRT will now process your raw data files and produce new BCDs, writing them into either your output directory or a subdirectory of your output directory called *bcd/* (depending on your platform).

29.6 Re-mosaic the data

Finally, re-make the input files as in Section 25.3, but this time remove all of the stim flash frames from the lists, and remove the first couple of frames from the list to mitigate the first-frame effect. Once you have edited the input lists, start a new Overlap flow in the MOPEX GUI by going to File > New Overlap Pipeline and selecting the MIPS 70 microns overlap template. Read in the input files as in Section 25.3,

and set a new output directory. Now add in the Mosaic pipeline by clicking on the yellow box near the top of your flow window labelled "Insert Mosaic..." and select the MIPS 70 microns template. This will append the Mosaicking pipeline to the end of the Overlap flow. Finally, hit the green "play" arrow in the top toolbar to set MOPEX going and create your final mosaic. The final product should look as follows:



You can download this as a FITS file here:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis tools/cookbook/files/mips70_stimcorr_stimcorr.fits

See the next recipe, entitled GeRT: Filter Extended Sources in MIPS-70 data, for instructions on how to filter this dataset to remove the vertical striping.

Recipe 30. GeRT: Filter Extended Sources in MIPS-70 data

This recipe describes how to use the GeRT to filter your MIPS 70 microns BCD frames for extended targets, e.g. galaxies. **WARNING: users who are interested in extended background emission such as the ISM should *not* filter their data, as this type of emission is removed during the process.**

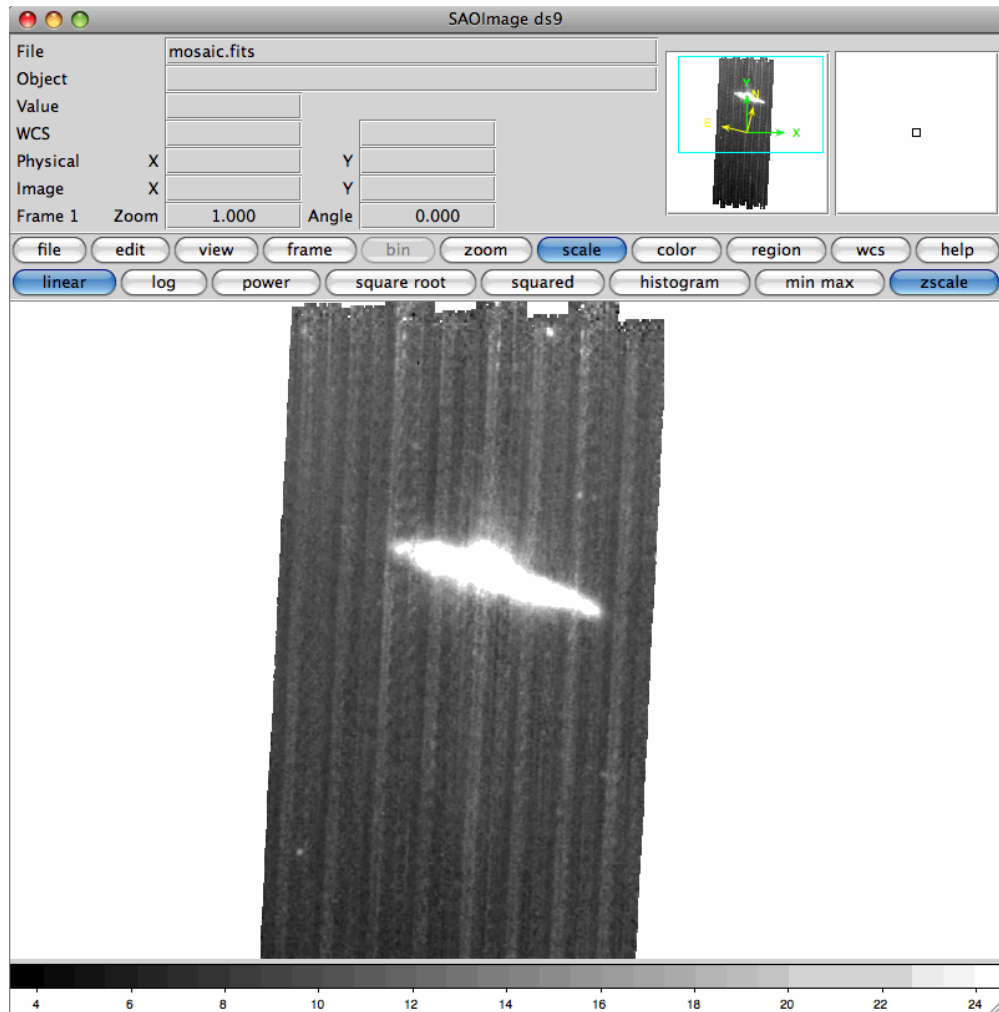
The filtered BCDs (**fbcd.fits*) downloaded from the archive and created automatically with the GeRT are optimised for point sources only. Users wishing to analyse data of extended sources must either work on the non-filtered BCD frames (**_bcd.fits*), or filter the BCDs with different parameters. This demonstration follows on from a previous recipe: GeRT: Bad Stim Flash Correction in MIPS-70 data, using the same data set. We assume that you have worked through the previous recipe before beginning this one.

30.1 Requirements

You will need to follow through the previous Data Analysis Recipe (Recipe 29) **GeRT: Bad Stim Flash Correction in MIPS-70 data** before beginning this demonstration.

30.2 Inspect the mosaic created in the previous Data Analysis Recipe

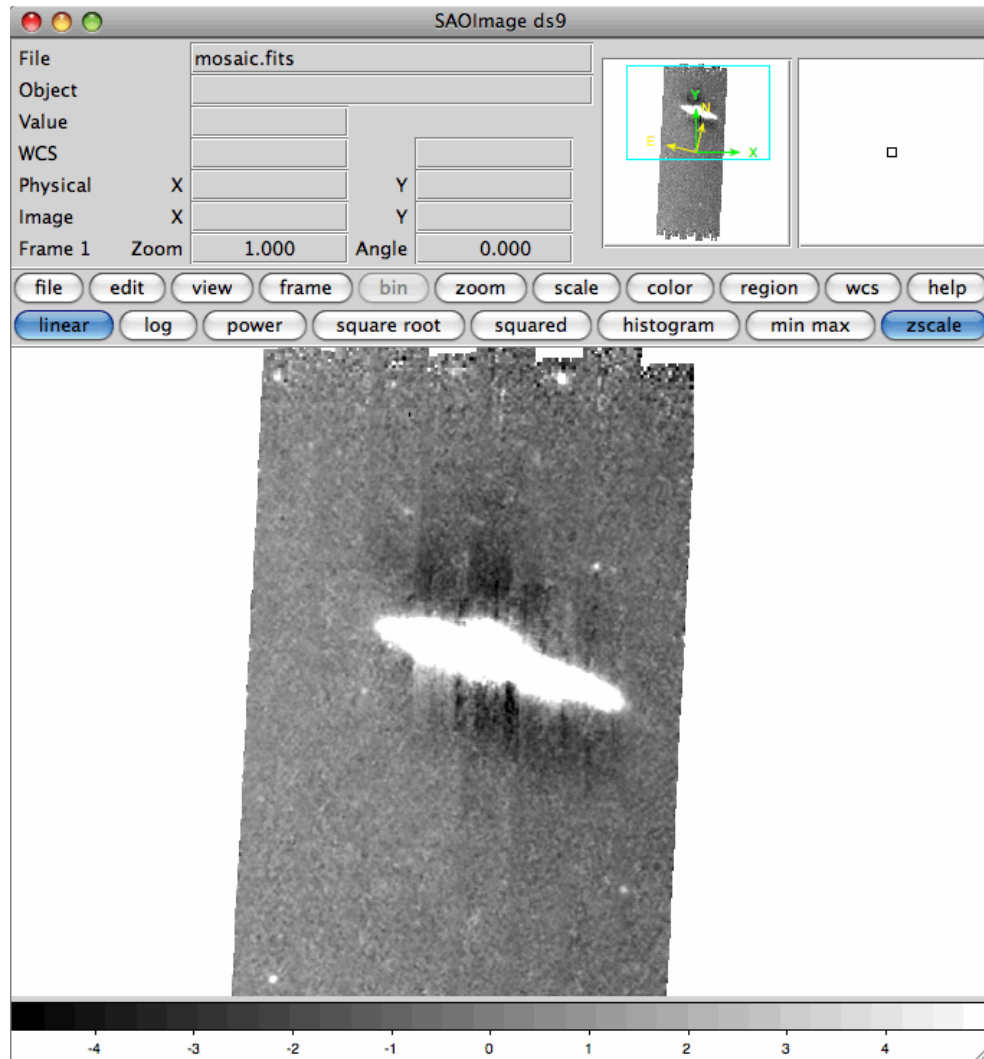
In the previous recipe, GeRT: Bad Stim Flash Correction in MIPS-70 data, you downloaded the MIPS 70 microns data from the archive, and corrected for a bad stim flash calibration. This left you with a mosaic that looked like this:



You can download this as a FITS image:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/mips70_stimcorr_stimcorr.fits

The next step in the reduction process is to filter out the vertical striping in the mosaic. When you download your data from the archive, or when you run the GeRT to reproduce the BCDs from the raw files, there are two types of Basic Calibrated Data frame generated: **bcd.fits* (the unfiltered BCDs) and **fbc.fits* (BCDs that have been filtered, with the filtering parameters optimised for point sources). Since we are looking at an extended source, we can't use the existing fBCDs. If you were to mosaic together the fBCD frames, or look at the filtered archive mosaic in the *pbcd/* directory (with the extension *.mfilt*) you would find the following, with negative side-lobes (dark areas) around the bright source:



You can download this as a FITS image:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/mips70_refilt_exps_fit.fits

In order to carry out an appropriate filtering for this extended source, we need to get an estimate of the position and the x and y sizes (in pixels) of the source. This will allow us to mask out the galaxy when calculating the median background level. Looking at the mosaic created in the previous recipe, we find that the galaxy is located at RA: 288.13909 degrees, Dec: 67.66165 degrees, approximately 33 pixels across and 105 pixels in length. This will become our mask position and size when running the GeRT.

30.3 Setting up the input files

In order to refilter the BCDs, the GeRT requires five input files:

- a list of the unfiltered bcds (*.bcd.fits);
- a list of the associated uncertainty files (*.bunc.fits);
- a list of the associated mask files (*.bmask.fits);
- an input namelist for masking out the bright galaxy when calculating the background level;
- an input coordinate list marking the center of the object(s) to be masked.

To create these files, change directory to the folder where the new BCDs were created in the last recipe:

```
unix% cd /where/you/unpacked_data/r5568512/ch2/raw/OUTdir/bcd/
```

Create the 3 lists of *.bcd.fits, *.bunc.fits and *.bmask.fits files from the command line as follows:

```
unix% ls *.bcd.fits > imagelist.txt
unix% ls *.bunc.fits > sigmalist.txt
unix% ls *.bmask.fits > masklist.txt
```

You need to ensure that all of the files you are working with were taken with the MIPS 70 microns detector set as the prime array. This information can be found in the image header with the keyword PRIMEARR = 1. Remove from the input lists any files that have PRIMEARR = 2 or PRIMEARR = 3.

Now you need to create an ASCII source table, *source.tbl* with the coordinates of the center of the object to be masked. The table must be in exactly the following format, giving the RA and Dec of the source in degrees. Please note that the vertical bars are essential, the values in the columns must fall between those bars, and the column headings are case-sensitive:

srcid	RA	Dec	
I	d	d	
1	190.524	32.542	

Finally, you need to set up the namelist file that controls the GeRT input parameters. This file is located in the subdirectory *cdf/* of your GeRT installation, and must stay there. Edit it in place, e.g. on my system I would go to:

```
unix% cd ~/Software/GeRT_Mac_070904/cdf
unix% emacs mask_pointsource_extended.nl
```

The lines you need to edit are:

```
PointSourceList =
'/where/you/unpacked_data/r5568512/ch2/raw/OUTdir/bcd/source.tbl',
MaskBox_Xsize = 180,
MaskBox_Ysize = 30,
```


where the dimensions of the MaskBox are slightly bigger than the size of the galaxy in the mosaicked image. The box size is specified in pixels.

30.4 Running the GeRT

Once the input files are set up, you can run the GeRT to refilter the BCDs. Change directory to the folder containing your data and ensure that the GeRT is ready to run by sourcing the setup file, e.g. for my system I would type the following:

```
unix% cd /where/you/unpacked_data/r5568512/ch2/raw/OUTdir/bcd/  
unix% source ~/Software/GeRT_Mac_070904/gert_setup.csh
```

The refiltering process is now run with the script *cleanup70_extended.tcsh* as follows:

```
unix% $WRAPDIR/cleanup70_extended.csh imagelist.txt masklist.txt  
sigmalist.txt OUTPUT
```

You will probably see a number of warnings and error messages flash by while the script is running. Usually these are normal and can be disregarded. Once the script has finished, move to the newly created output directory (named as you specified on the command line, but with a prefix "cc" e.g. *ccOUTPUT*) and check whether the new BCDs have been created. If you have created the output directory as a subdirectory of your current working directory then the files will be in a further subdirectory called *bcd*, i.e. in *ccOUTPUT/bcd*. If you specified that the output directory is made elsewhere then the BCD files will be in the top-level of the output directory (i.e. in *ccOUTPUT*). If the new files exist then everything worked. These files are your re-filtered BCDs, optimised for the analysis of extended sources.

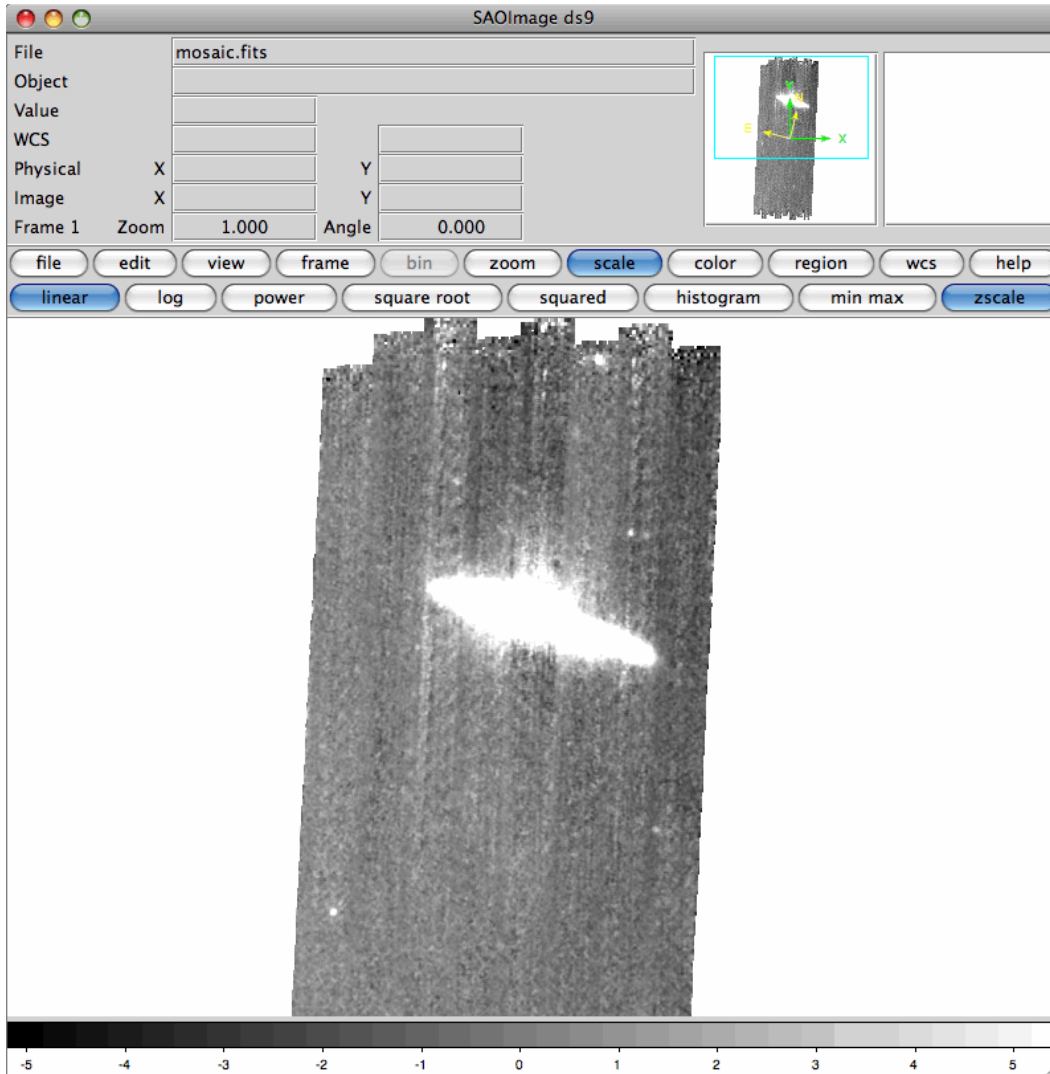
30.5 Re-Mosaicking the new BCDs

The final step is to use MOPEX to re-mosaic the re-filtered BCDs to check the results of the refiltering. Re-make the input lists *imagelist.txt*, *sigmalist.txt* and *masklist.txt* as in Section 26.3, but this time list the new BCDs in *imagelist.txt*, remove all of the stim flash frames from the three lists (i.e. files with header keyword *STMFL_70 = 1*), and remove the first couple of frames from them to mitigate the first-frame effect.

Once you have created the input lists, start the MOPEX GUI and load the Overlap template for MIPS 70 microns data (go to File > New Overlap Pipeline > Overlap, 70 microns and hit "ok"). Now insert the Mosaic pipeline by clicking on the "Insert Mosaic" button at the top of the flow. Set up the flow by setting the Image Stack File to point to the new list of BCDs and choosing an Output Directory. Be patient when trying to load the Image Stack File - the directory contains a lot of images, and it takes a while (up to a minute) for MOPEX to load the directory contents into the file selection window. Next you need to click on "Optional Input and Mask Files" and set the Sigma List File and the DCE Status Mask

File to point to your *sigalist.txt* and *masklist.txt* files, respectively. Again, be patient as it takes a while for MOPEX to display the directory contents. The Pmask FITS File should be set to the MIPS 70 microns PMask file found in the *cal/* directory of your MOPEX installation (*cal/mips70_pmask.fits*) but the remaining input files can be left blank. Click "OK" and then start the flow running by clicking on the green "play" arrow in the top-left corner of the MOPEX window. Again, be patient when waiting for the flow to start.

Once the flow has finished, you should have a mosaic that looks like the one below.



You can download this as a FITS image:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/mips70_refilt_ex_final.fits

While the correction to the striping isn't perfect, this is often as good as it is possible to get when a bright object is near the end of a scan leg. By removing the poorly calibrated stim flashes that were taken on top of the galaxy (see Recipe 29), there are often no more stim flashes before the end of the scan legs, and so

the calibration is extrapolated from the stim flashes taken earlier. This can lead to imperfect calibration at the end of the scan leg. Despite this, the effect is small - a tiny fraction compared to the bright source.

Recipe 31. GeRT: Correct Mild Saturation in MIPS-70 data

This recipe describes how to use the GeRT to recover mildly saturated sources in your MIPS-70 data. MIPS BCDs are created from the raw data cubes by calculating the slopes of the data ramps for each pixel. For more information about this, see the MIPS Data Handbook and Gordon et al. 2005 (PASP, 117, 503). Normally the BCDs use at least 4 samples to calculate the slope of the ramp, but if the pixel in question saturates before the last of these 4 samples, the pixel is flagged and masked in the BCD frame. The GeRT can be adjusted to re-make the BCDs using fewer samples, so, for example, pixels that are saturated at 4 samples, but not at 3, are recovered. The GeRT can generate BCDs using as few as 2 samples.

WARNING: changing the number of samples used to calculate the slope of the data ramp will affect the calibration of the data. You must read the 70 micron calibration paper (Gordon et al. 2007, PASP, 119, 1019) if you intend to use the GeRT to recover saturated pixels in your scientific data.

Obviously there are limits to the correction - pixels that are saturated before the second sample in the data ramps cannot be recovered, and so the GeRT can only work in cases of mild saturation. Here we describe how to re-create the BCDs using only 2 data samples to calculate the slope of the data ramp, so recovering the flux information from pixels that are mildly saturated in the postBCD image.

31.1 Requirements

You must have MOPEX (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/mopex/>) and the GeRT (<http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/tools/gert/>) installed in order to follow along with this recipe.

You can download the example data for this recipe from the SSC web site: (http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysisitools/cookbook/files/mips70_saturation_recipe_data.tar). You will need 435 MB of free disk space in order to follow along with this recipe.

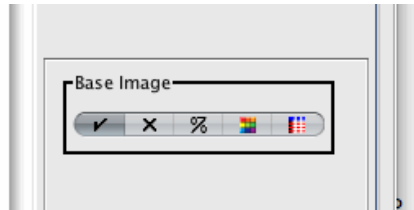
31.2 Inspect the Post-BCD mosaic to check for saturation

Download the example data via the link above, move the tarball into a clean working directory and unpack it. This should create 2 subdirectories labelled *pbcd/* and *raw/* containing the post-BCD mosaics and associated raw files from the Spitzer archive. The raw files provided in the tarball are a subset of the entire mosaic (to cut down on required disk space), so the final mosaic you create with these data will be smaller than the mosaicked image in the *pbcd* directory.

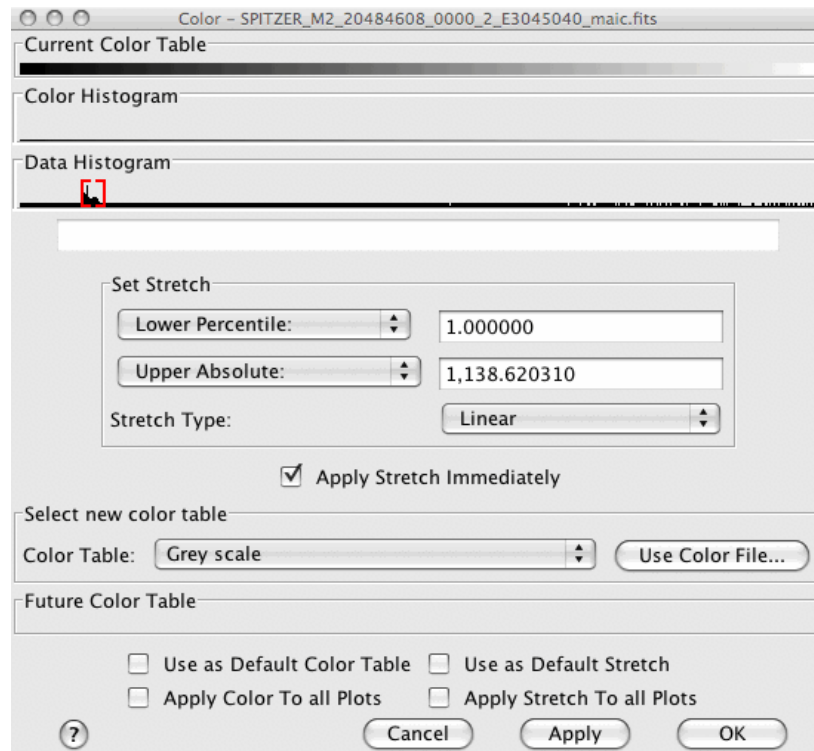
```
unix% tar xvf mips70_saturated_corr.tar
```

Open the MOPEX GUI and open the post-BCD mosaic, called *pbcd/SPITZER_M2_20484608_0000_2_E3045040_maic.fits* to see the final archive product (go to Image > FITS File Image). You can also do this in the image viewer of your choice (e.g. SAOImage DS9), but be aware that pixel NaN values (which show the position of the saturated data) show up as white in many viewers, rather than black (as in MOPEX) and so will not be immediately apparent.

First we need to re-scale the image to see the saturated areas. Click on the little color palette in the "Base Image" control bar on the right hand side (4th button from left):



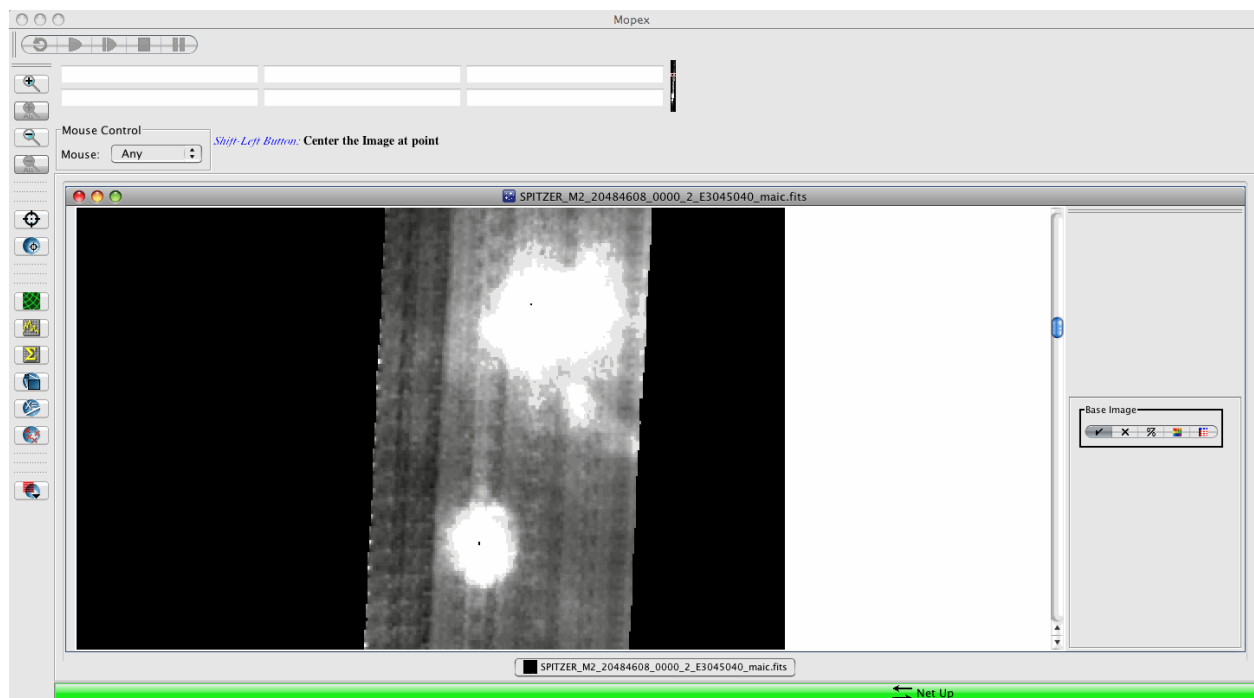
This will bring up a dialog box. In the "Data Histogram" slider, drag the right-hand red bracket towards the left-hand side until you start to see the background detail appearing in the image.



The MOPEX window should now look as follows:



Now we need to find the saturated areas of the image. Scroll down the image to find the following two areas of extended emission. This image is zoomed in one level to show the saturated spots more clearly:



The small black spots in the areas of extended emission are areas of saturation. If you run your cursor over them, you will see that the flux value shown in the top left-hand corner of MOPEX disappears for those pixels (NaNs in the data). We will use the GeRT to re-make the BCDs to correct for these.

31.3 Setting up the GeRT input files

The GeRT needs two input files to recreate the BCDs from the raw data cubes. Firstly, we need to create a list of the raw data cubes to feed into the GeRT. Change into the *raw/* directory and list all of the FITS files into an ASCII file:

```
unix% cd raw
unix% ls *raw.fits > rawlist.txt
```

Next we need to set up the GeRT to use only 2 data samples rather than the standard 4 to calculate the slopes of the data ramps. To do this we need to edit the file called *MIPS70_SLOPE_0.nl* found in the *cdf/2/* directory of your GeRT installation:

```
unix% cd /your/GeRT/installation/directory/cdf/2/
unix% emacs MIPS70_SLOPE_0.nl
```

Scroll down the file until you find the parameter block starting with

```
&SLOPE
```

and change the following parameter from:

```
Min_Num_Samples = 4,
```

to

```
Min_Num_Samples = 2,
```

Important: your edits must include a space either side of the "=" sign, and must include the comma at the end of the line, else your changes will not be read.

In addition, we need to change the &CVTIN block to tell the GeRT to use all of the data, rather than rejecting the first frame in the data ramp:

```
&CVTIN
  DataHi = 0,
  DataLo = 0,
```

```
SatHi = 65500,  
SatLo = 10,  
StimHi = 4,  
StimLo = 4,  
&end
```

Once you have made these changes, save the file.

If this is the first time that you have used the GeRT, follow the setup instructions before going any further. The instructions can also be found in the GeRT installation directory (file called README). Once the GeRT is set up, return to your working directory *raw/*.

```
unix% cd /where/you/unpacked_data/raw/
```

31.4 Running the GeRT

Once the input files are set up, you can run the GeRT to recreate the BCDs. From the directory containing the raw data, ensure that the GeRT is ready to run by sourcing the setup file, e.g.:

```
unix% source /your/GeRT/installation/directory/gert_setup.csh
```

If you see any error messages then check the GeRT setup instructions to make sure that you have set the GeRT up correctly.

Run the GeRT to re-create the BCDs as follows:

```
unix% $WRAPDIR/gert.pl rawlist.txt ../OUTdir
```

where *rawlist.txt* is the list of raw files you created earlier, and *../OUTdir/* is the output directory. I chose to create the directory on the same level as the *raw/* directory rather than as a subdirectory. This is not necessary, but comes in useful when trying to run MOPEX, as the file selection dialogue box in MOPEX takes a long time to show subdirectories if the directories contain a lot of files. The output from the GeRT should look something like this:

```
GeRT S14.0 v1.1 with 0 threads  
0001 <* SPITZER_M2_20484608_0000_0033_2_raw.fits  
0002 : SPITZER_M2_20484608_0000_0034_2_raw.fits  
0003 : SPITZER_M2_20484608_0000_0035_2_raw.fits  
0004 : SPITZER_M2_20484608_0000_0036_2_raw.fits  
.....  
.....  
0291 : SPITZER_M2_20484608_0003_0222_2_raw.fits
```



```
0292 : SPITZER_M2_20484608_0003_0223_2_raw.fits
0293 : SPITZER_M2_20484608_0003_0224_2_raw.fits
SLOPER Processing done. 293 dce's in 273 seconds.
CALER Processing done, BCDs in ../OUTdir/bcd processed in 29 seconds.
Total processing of 293 dce's processed in 302 seconds
```

Check to see that the BCDs have been created in the output directory:

```
unix% cd ../OUTdir
unix% ls
```

The directory listing should show a subdirectory for each processed BCD, alongside the final recreated BCD image files and associated filtered BCDs, masks and uncertainty files. e.g.:

```
unix% ls *.fits

OUTdir.0001.bcd.fits
OUTdir.0001.bmask.fits
OUTdir.0001.bunc.fits
OUTdir.0001.fbcd.fits
OUTdir.0002.bcd.fits
OUTdir.0002.bmask.fits
OUTdir.0002.bunc.fits
OUTdir.0002.fbcd.fits
OUTdir.0003.bcd.fits
OUTdir.0003.bmask.fits
OUTdir.0003.bunc.fits
OUTdir.0003.fbcd.fits
.....
```

31.5 Re-Mosaicking the new BCDs

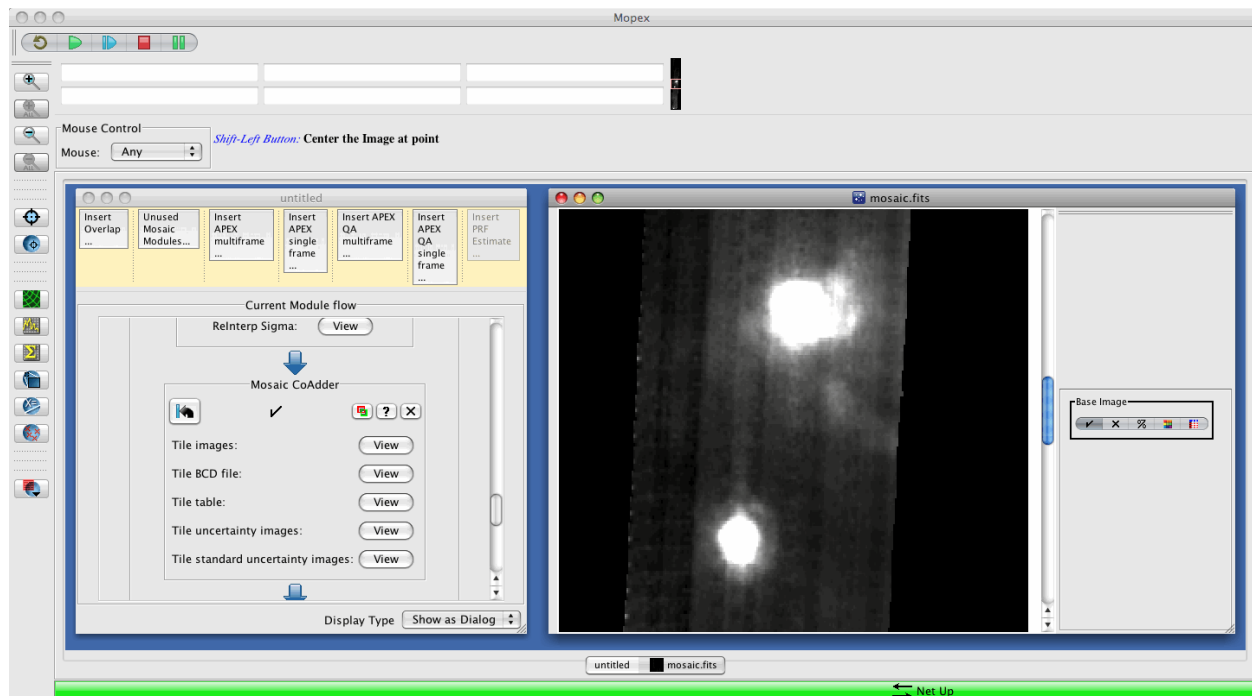
The final step is to use MOPEX to re-mosaic the new BCDs to check the results of the saturation correction. First we must create the input lists of images, associated masks and associated uncertainty files. List all of the **bcd.fits*, **bunc.fits* and **bmask.fits* frames for input as follows:

```
unix% ls *.bcd.fits > imagelist.txt
unix% ls *.bunc.fits > sigmalist.txt
unix% ls *.bmask.fits > masklist.txt
```

Now remove all of the stim flash frames (all frames with header keyword STMFL70 = 1 are stim flash frames and should be rejected), and check that all of the data frames were taken with the 70 micron array as the prime array (PRIMEARR = 1; reject any frames with PRIMEARR = 2 or PRIMEARR = 3).

Once you have created the input lists, start the MOPEX GUI and load the Mosaic template for MIPS-70 data (go to File > New Mosaic Pipeline > Mosaic, 70 microns and hit "ok"). Set up the flow by setting the Image Stack File to point to the new list of BCDs, *imagelist.txt*, and choosing an Output Directory. Be patient when trying to load the Image Stack File - the directory contains a lot of images, and it can take a while for MOPEX to load the directory contents into the file selection window. Next you need to click on "Optional Input and Mask Files" and set the Sigma List File and the DCE Status Mask File to point to your *sigmalist.txt* and *masklist.txt* files, respectively. Again, be patient as it takes a while for MOPEX to display the directory contents. The Pmask FITS File should be set to the MIPS-70 PMask file found in the *cal/* directory of your MOPEX installation (*/your/mopex/directory/cal/mips70_pmask.fits*) but the remaining input files can be left blank. Click "OK" and then start the flow running by clicking on the green "play" arrow in the top-left corner of the MOPEX window. Again, be patient when waiting for the flow to start.

Once the flow has finished, you should have a mosaic that looks like the one below, where you can see that the previously saturated areas of the extended emission regions now look good. **REMEMBER - following this procedure will have affected the calibration of your data. Do not publish data processed in this way without first reading the MIPS-70 calibration paper (Gordon et al. 2007, PASP, 119, 1019).**



You can download this as a FITS image:

http://irsa.ipac.caltech.edu/data/SPITZER/docs/dataanalysis/tools/cookbook/files/mips70_saturation_final.fits

Obviously, the background for this image is far from perfect, and additional processing will need to be done in order to correct for bad stim flash calibration and to refilter the BCDs to optimise them for photometry on extended objects. See the recipes “GeRT: Bad Stim Flash Correction in MIPS-70 data” and “GeRT: Filter Extended Sources in MIPS-70 data” for step-by-step instructions on how to do this. Refiltering the data to optimise for extended sources is only recommended if you are not interested in diffuse background emission such as the ISM.

Appendix A. Version Log

Version 6.0, June 2017:

Several edits were made to Recipes 1-7

Version 5.0.1, October 2012:

Fixed broken links

Version 5.0, January 2012:

Updated Recipes 1 and 2 for the Spitzer Heritage Archive v 3.0

Version 4.0.1, June 2011:

Fixed broken links

Version 4.0, April 2011:

Removed obsolete recipe "Downloading Data from the Spitzer Archive using Leopard".

Version 3.0, February 2011:

Updated Recipe 7, "MOPEX/APEX: Point Source Photometry from IRAC Images"

Version 2.0, September 2010:

Two recipes have been added to describe the use of the Spitzer Heritage Archive (Recipe 1 and Recipe 2).

Updated Recipe 3 ("COSMOS IRAC channel 1 Mosaic") to include advice on what would change for a shallower or deeper survey. This recipe also now includes a warning about how drizzle modifies the noise properties of an image in a non-linear manner. Finally, the recipe includes an update about the new background subtraction algorithm implemented in the medfilter module in MOPEX 18.2.

Added Recipe 6, "Making an IRAC Array Location Dependent Correction Mosaic"

Added Recipe 7, "MOPEX/APEX: Point Source Photometry from IRAC Images"

Added Recipe 18, "IRSCLEAN, COAD, and SPICE: Mrk 273 LH Data"

SMART recipes have been updated (Recipe 19 and Recipe 20).