



NHSC/PACS Web Tutorials

Running PACS photometer pipelines

PACS-402 (for Hipe 12.0)

*Level 1 to Level 2 processing:
The JScanam pipeline*

Prepared by Roberta Paladini
(with inputs from Javier Gracia-Carpio & Michael Wetzstein)

June 2014

PACS-402

1



Level 1 to Level 2 processing:

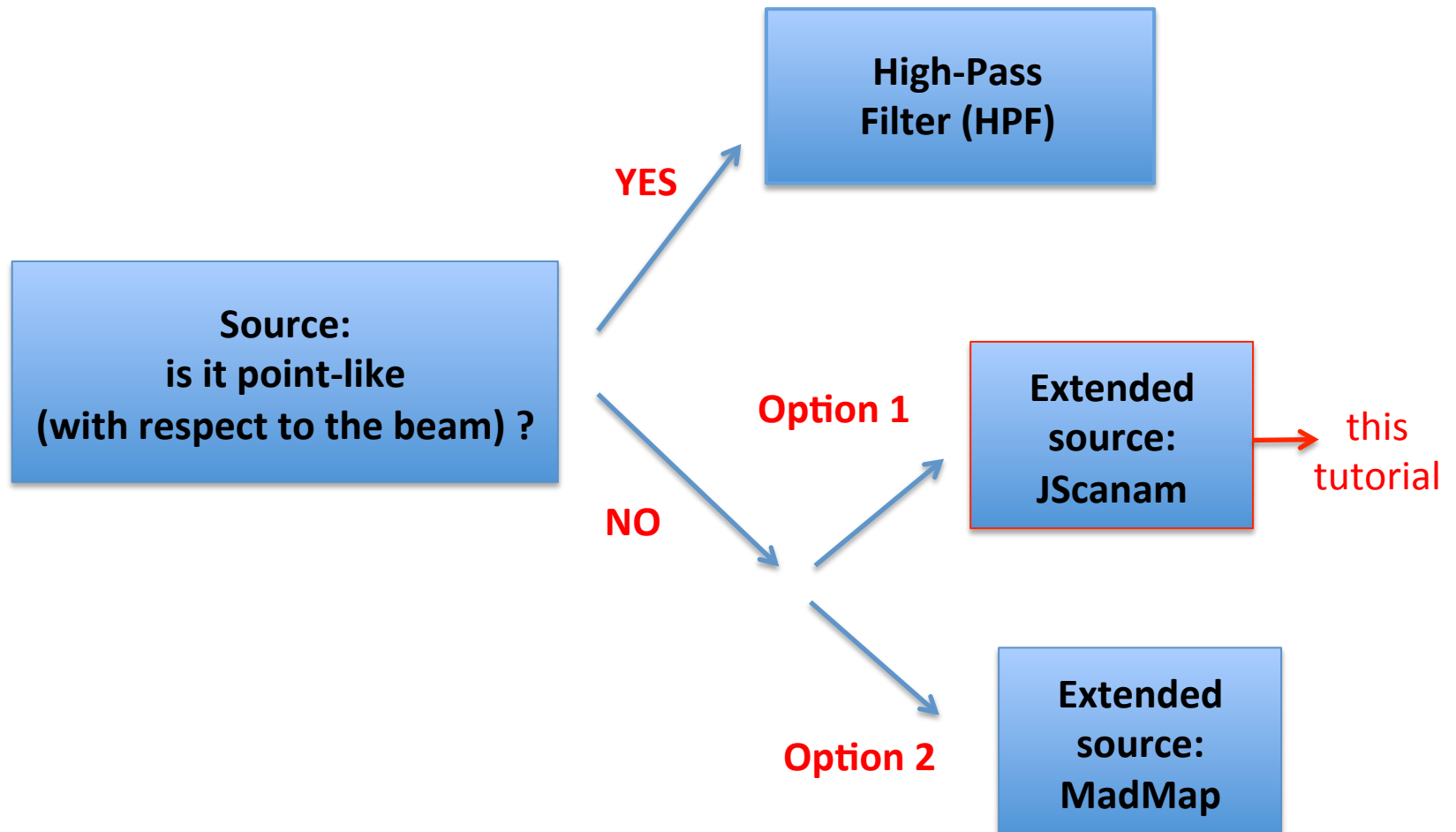
NOTE: The pipeline is now very stable between Level 0 and Level 1, so there is no need to tweak the processing between the raw-data level (i.e. Level 0) and Level 1.



Outline of this tutorial

- Slide 5 to 23: Jscanam philosophy and walk-through the ipipe script
- Slide 24: useful notes
- Slide 25: documentation

PACS Photometer Pipeline: 2 main branches





What is JScanam ?

- As pointed out in the previous slide, JScanam allows the processing of PACS photometer data when the user wants to preserve extended emission
- JScanam is the Java port in HIPE of IDL Scanamorphos (Roussel et al. 2013)
- For detailed information on individual steps of the processing, we refer the reader to the Scanamorphos web page:

<http://www2.iap.fr/users/roussel/herschel/>



Where is the JScanam ipipe script ?

The screenshot shows a PyCharm IDE window titled "HIPE - /Users/paladini/Documents/PACS/HIPE/hcss.dp.pacs-12.0.2749/scripts/pacs/scripts/ipipe/phot/scanmap_Extended_emission_JyPScanam.py". The editor displays the following Python code:

```
89 """
90 camera = "red"
91 obsids = [1342211283, 1342211284]
92 solarSystemObject = False
93 galactic = True
94 calculateRaDec = False
95 showMapsAfterTasks = False
96 debug = False
97 declitch = True
98 nSigmaDeclitch = 3
99 makeFinalMap = True
100 outputPixelSize = -1
101 pixfrac = 0.1
102
103 """
104 ##### SECTION 1 #####
105 ##### PROCESSING #####
106 """
107
108 print "-- Executing JScanam -- "
109
110 """
111 Load Level 1 data from the HSA
112 """
113 print " Loading Level 1 data "
114 scansObs = getObservation(obsids[0], useHsa=True, instrument="PACS")
115 level1 = PacsContext(scansObs.level1)
116 scans = level1.averaged.getCamera(camera).product.selectAll()
117 blueFilter1 = scansObs.meta["blue"] value
```

The console window at the bottom shows the following output:

```
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
```

The status bar at the bottom indicates the user is "rpaladin" using the "Jython Interpreter 100%" with 4892 of 10775 MB of memory used.



The JScanam ipipe script in a nutshell

1. turnarounds removal
2. mask long-term glitches
3. baseline subtraction after masking the sources
4. destriping, i.e. removal of the signal drift on time scales **longer** than a scan leg
5. individual drift removal, i.e. removal of the drift on time scales **shorter** than a scan leg
6. creation of the final map



.. a note about: baseline and drifts

Flux Calibration Offsets: small voltage variations (i.e. offsets) from bolometer to bolometer. The net effect of these offsets are *gradients* which can be corrected for by removing 1st-order polynomials (or *baselines*)

Long-Term Thermal drift: low-frequency noise component (e.g. 1/f) due to small thermal fluctuations of the bolometers cryogenic bath. This component is correlated across the PACS bolometers

Short-Term Drift: low-frequency noise component (e.g. 1/f) of non-thermal nature. It is due to detector physics and readout components. It is uncorrelated across bolometers



Let's take a closer look at the script....

If you want the script to output some diagnostic plots, at the beginning set:

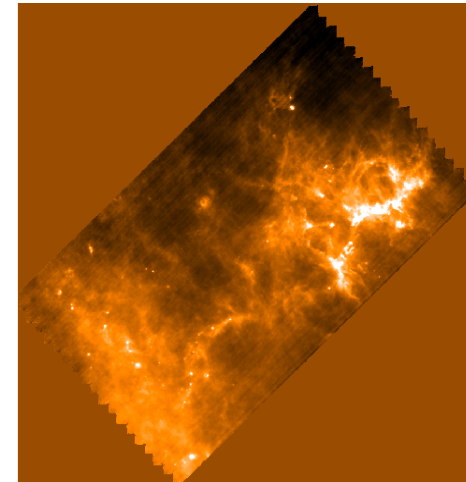
debug = True

NOTE: make sure that you use the “True” option only for debugging purposes. With this option, the script in fact will generate MANY, MANY plots...

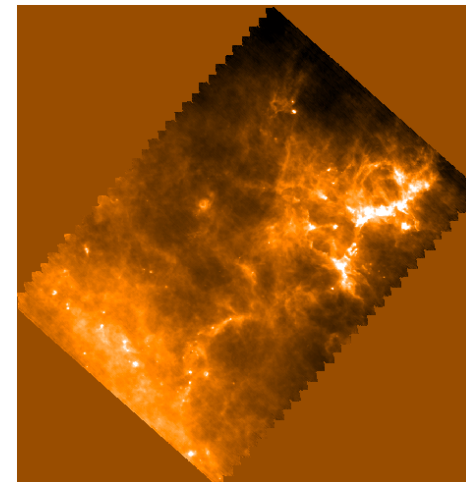
Test Case: summary

- ✓ We use the pair of OBSIDs:
1342204362/1342204363
- ✓ This is M16, a Galactic star forming region (→ bright, structured background)
- ✓ This is a PACS/SPIRE parallel mode observation
- ✓ The data set consists of a single set of orthogonal scan & cross-scan observations
- ✓ Both scan and cross-scan cover an area in the sky: 30' X 60'

NOTE: in the following slides, we will only show the results for the scan direction. Very similar results are obtained for the cross-scan.



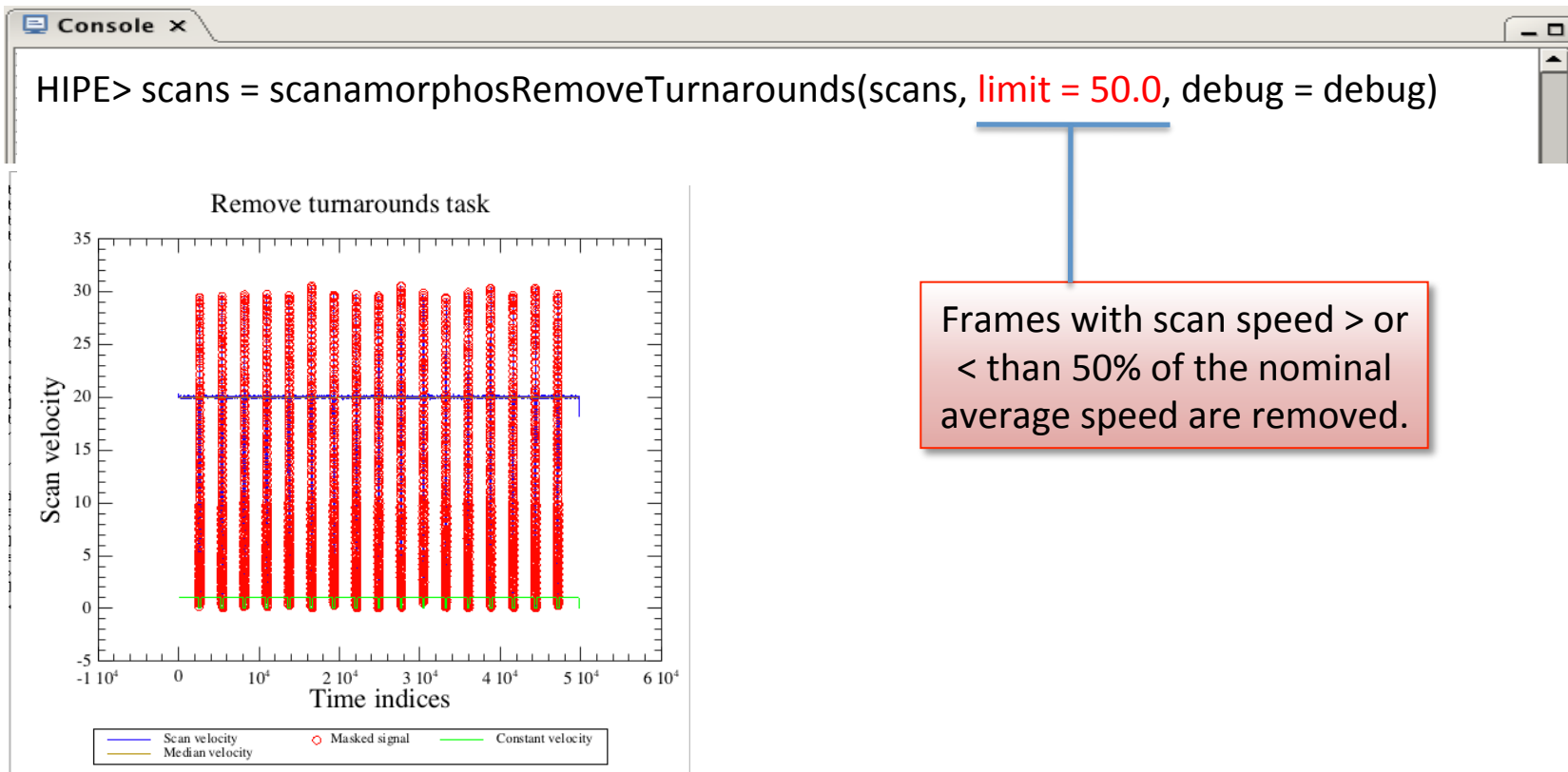
Scan map



Cross-scan map

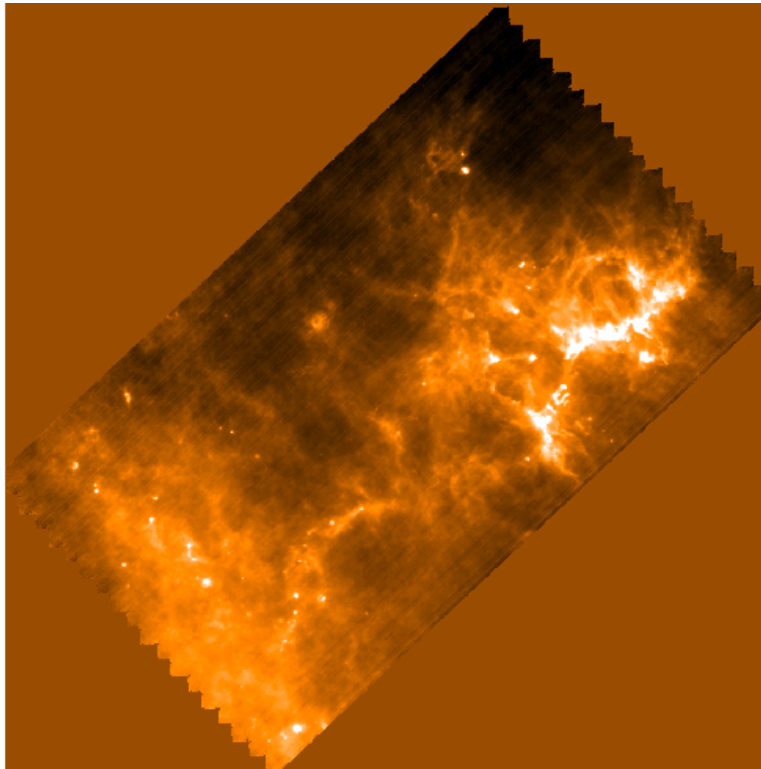
1. Turnaround Removal

The frames which are taken during the turnaround of the spacecraft are removed. These frames are identified based on the speed scan: frames with speed $>$ or $<$ than TOT % of nominal average speed are removed.



2. Mask long-term glitches

```
Console x  
HIPE> scans = scanamorphosMaskLongTermGlitches(scans, stepAfter = 20, galactic =  
True)
```



Scan map after long-term glitch removal

NOTE: The task generates a glitch mask (*Scanam_LongTermGlitchMask*). It is advisable to check this mask. In case some glitches are not detected, one can improve the performance of the task by twiking the *stepAfter* parameter



3. Baseline Subtraction

The baseline subtraction is a multi-step process:

- subtract a linear baseline from the signal of every pixel and scan leg: the goal is to obtain a first-order map which allows the creation of a source mask
(→ *scanamorphosScanlegBselineFitPerPixel*)
- create a source mask and attach it to the original scans (→ *scanamorphosCreateSourceMask*)
- now remove general offset: 1st pass (→ *scanamorphosBaselineSubtraction*)
- identify and remove calibration block (→ *scanamorphosBaselinePreprocessing*)
- now remove real baseline: 2nd pass (→ *scanamorphosBaselineSubtraction*)

Source Masking

Before correcting the data for the baseline, we need to mask both extended and compact sources. This prevents the offsets estimate and removal to be minimally affected by the presence of sources.

```
Console x  
HIPE> sourceImage, scans = scanamorphosCreateSourceMask(scans, nSigma = 4.0,  
createMask = False, galactic = True, calTree = calTree, debug = debug)
```

`createMask` is set to True when the mask is applied to the frames



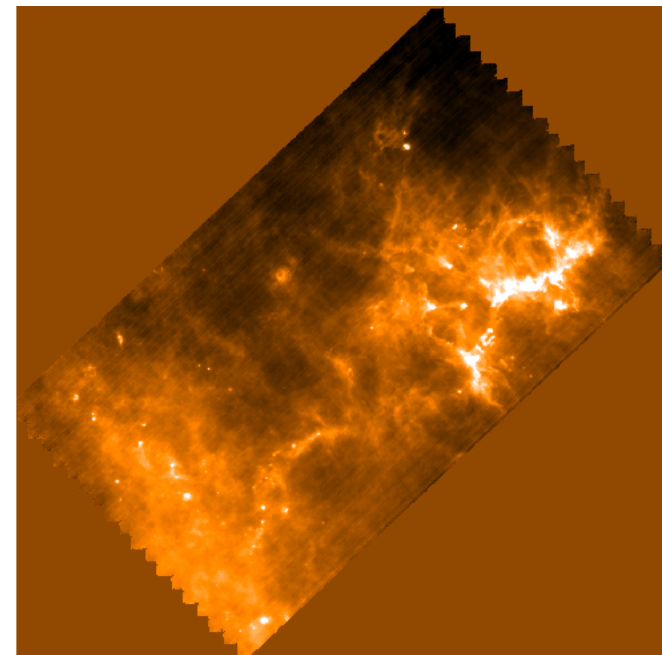
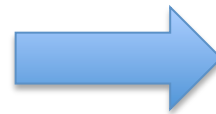
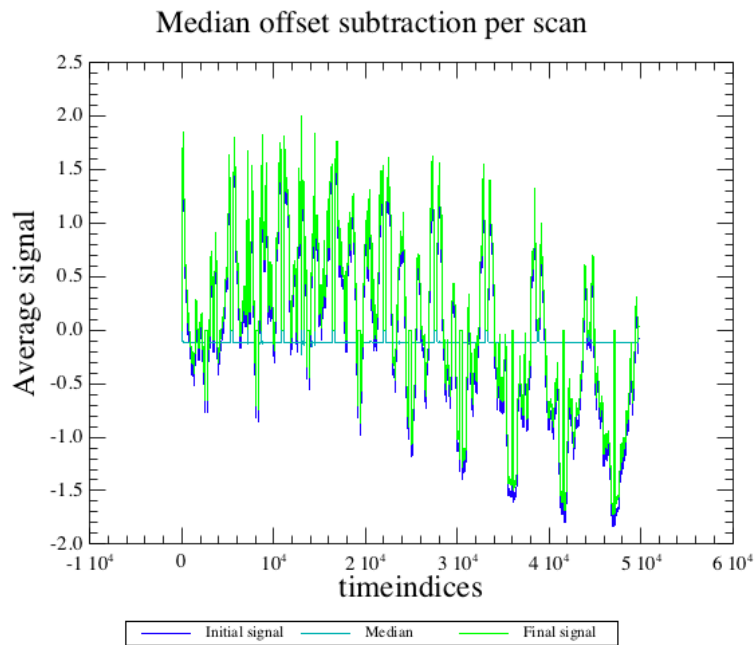
Source mask

nSigma controls which emission should be masked.
NOTE: the mask should cover only < 30% of the map

Baseline Removal: 1st pass

Using only the unmasked values, the **signal median value** in every pixel array and every scan (**galactic = True**) or scan leg (**galactic = False**) is computed and **subtracted from the frames**.

```
Console x  
HIPE> scans = scanamorphosBaselineSubtraction(scans, galactic = True, debug = debug)
```



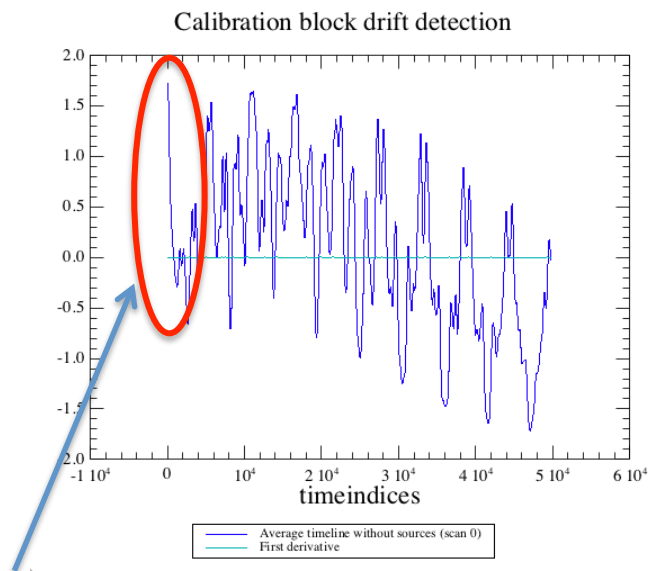
Scans map after baseline subtraction: 1st pass

Calibration Block Removal

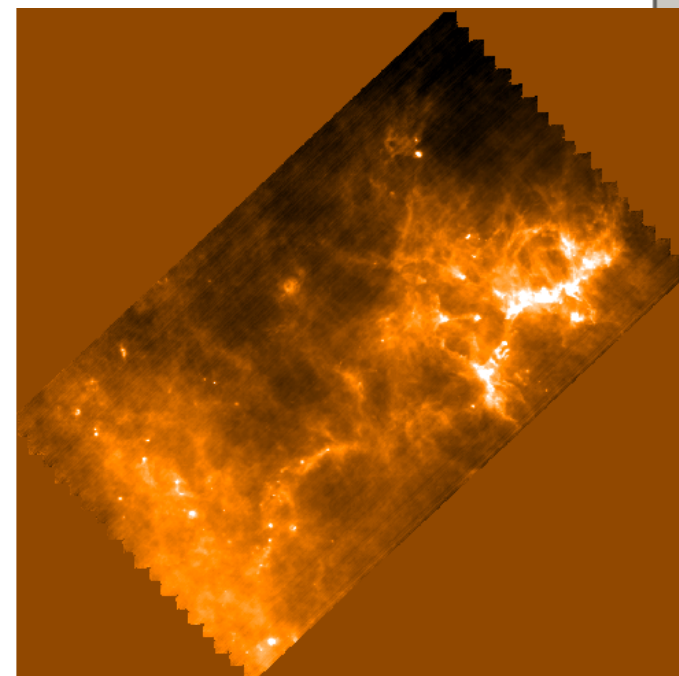
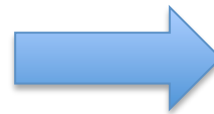


A calibration block is scheduled at the beginning of each PACS photometer observation. It is known to cause a strong signal drift which affects the first ~1000 frames. These are identified and masked.

```
Console x  
HIPE> scans = scanamorphosBaselinePreprocessing(scans, galactic = True, debug =  
debug)
```



Frames affected by calibration block

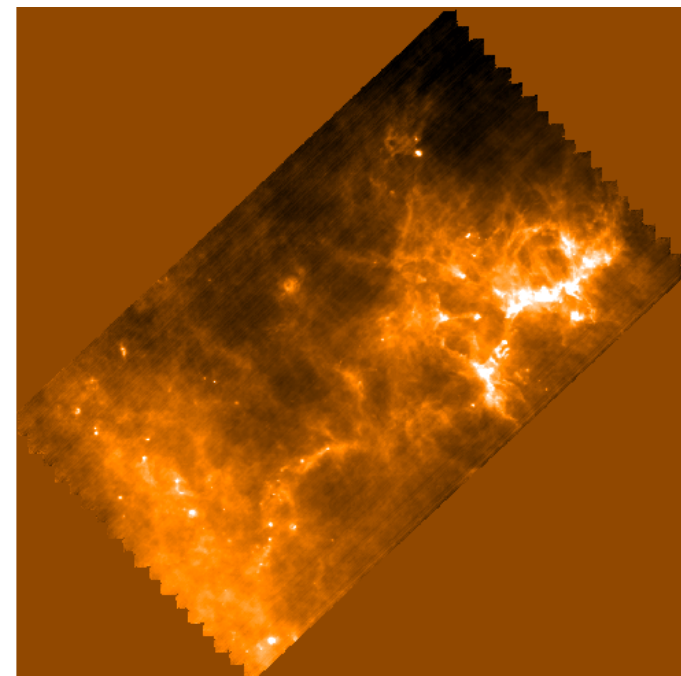
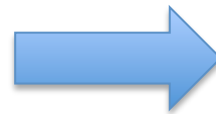
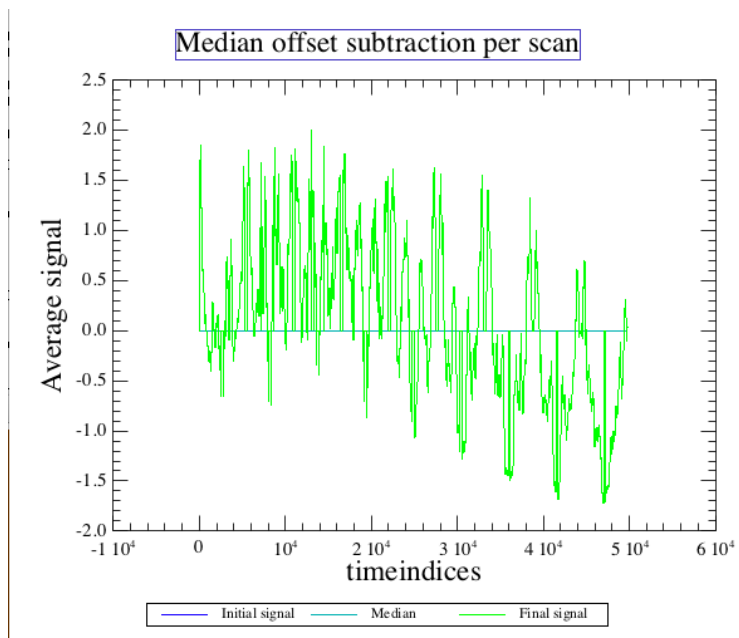


Scans map after removal of calibration block

Baseline removal: 2nd pass

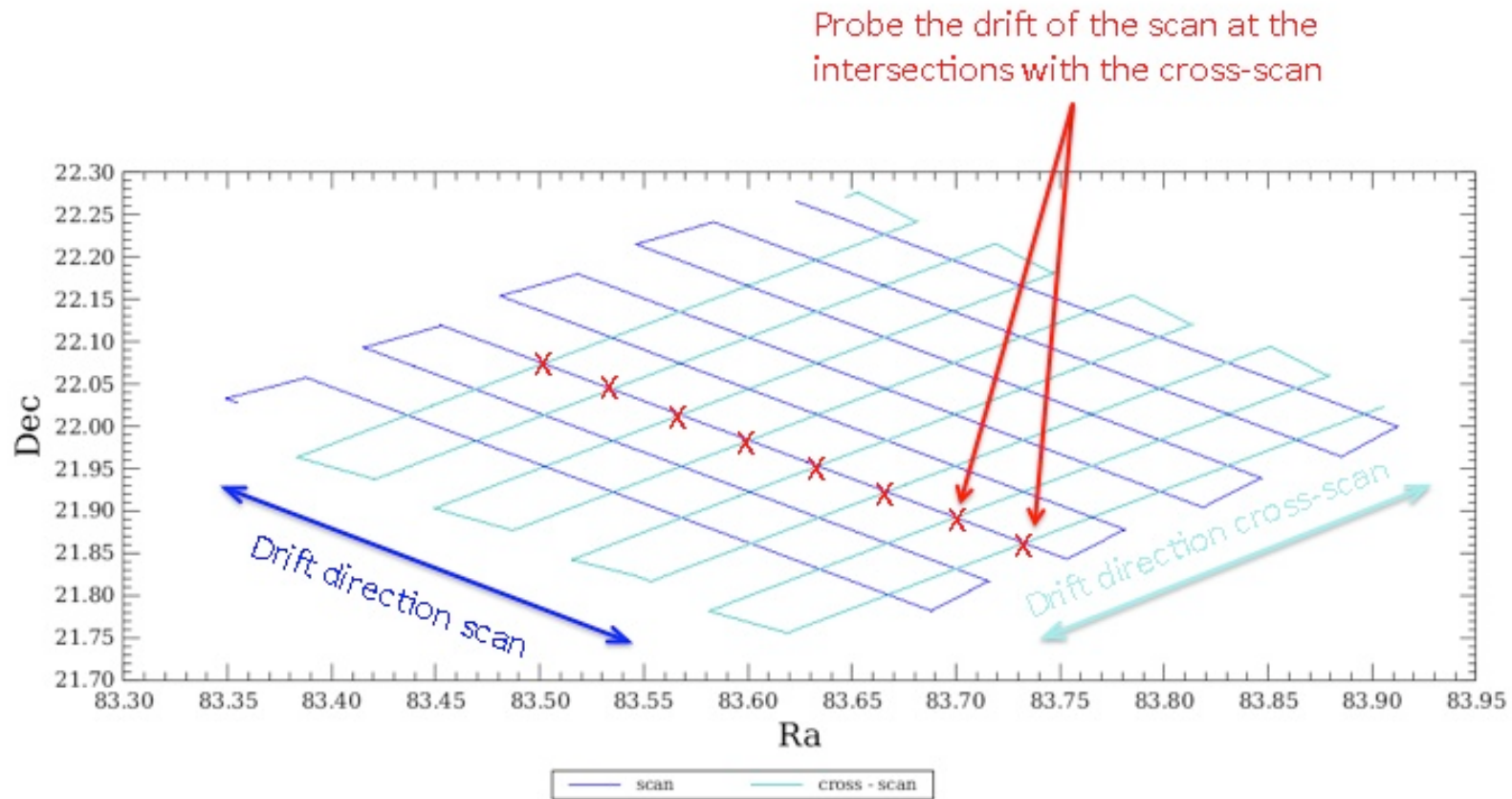
After the removal of the calibration block, repeat the baseline subtraction.

```
HIPE> scans = scanamorphosBaselineSubtraction(scans, galactic = True, debug = debug)
```



4. Destriping: concept

The basic idea is to remove the remaining offsets and long-term drift by comparing the value of a scan-pixel timeline with the value of the cross-scan timeline at that same position of the sky, i.e. where the scan and cross-scan timeline cross.



How destriping works

```
Console x
HIPE> scan, cscans = scanamorphosDestriping( scans, cscans, iterations = 6, calTree=
calTree, debug = debug)
```

This is a multi-step process:

[\(all the steps are performed by the same task in this case\)](#)

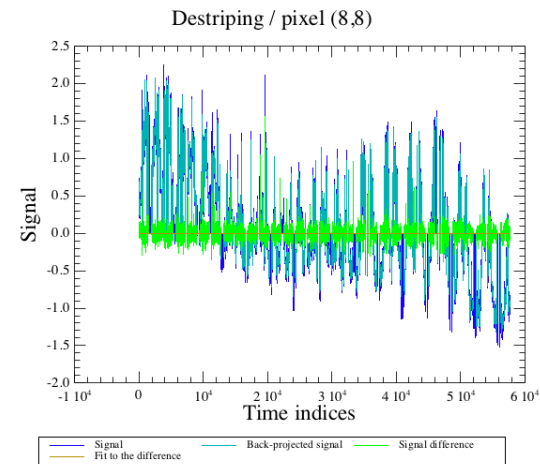
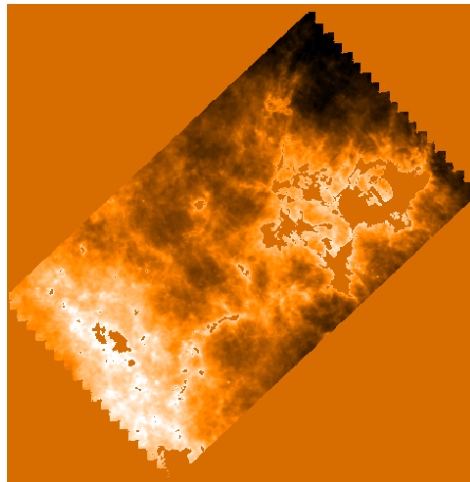
- create a map of the cross-scans and back-project it to the scan direction of the scans
- subtract the back-projected timelines from the scans timelines
- fit a line to the timeline difference and subtract it from the scan lines
- repeat all previous steps on the cross-scans, using a back-projection of the scans
- repeat until convergence is reached, i.e. when the fit of the difference is smaller than the fit residuals

NOTE: Because we are dealing with $1/f$, the longer the time scale (i.e. the frequency f), the more the drift. Therefore, the destriping is performed by the task on different time scales, that is: 1) the whole scan duration; 2) four scan legs; 3) one scan leg.

Ex: one iteration

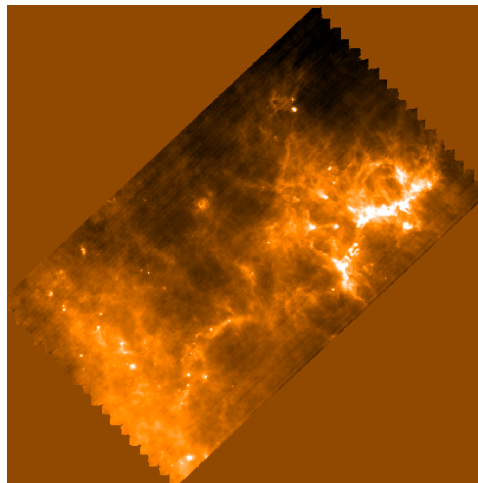


Reference scan map:
back-project it to cross-scan direction.
Mask the sources to avoid affecting long-term drift correction

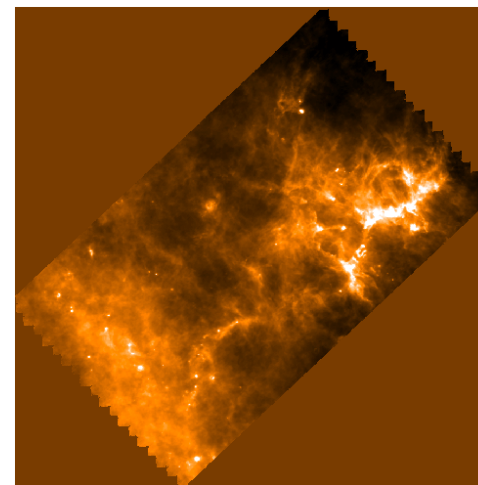
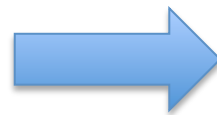


Subtract scan back-projected timelines from the cross-scan timelines. Fit the difference and subtract it from the cross-scan timelines

Destriping: final results



Scan map before destriping



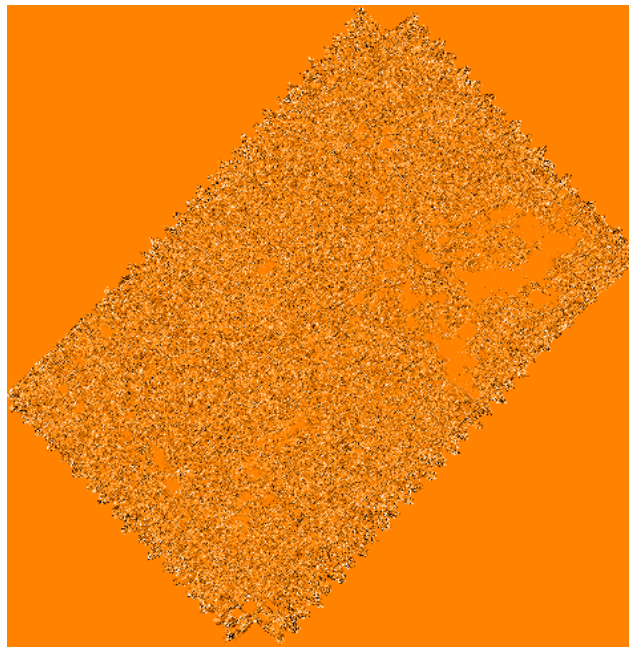
Scan map after destriping

PACS-402

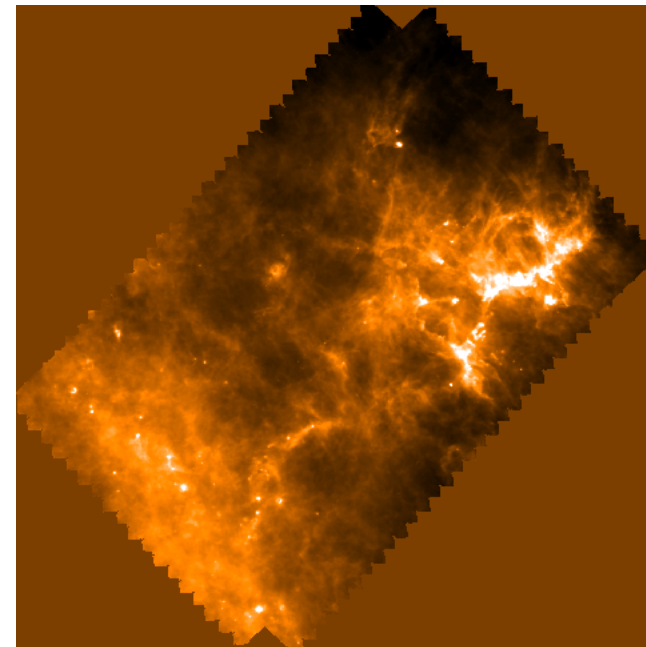
5. Individual drift removal

```
Console x  
HIPE> mergedScans = scanamorphosIndividualDrifts(mergedScans, calTree = calTree,  
debug = debug)
```

NOTE: that scan and cross-scans are merged before applying this task



Individual drift map



Scan and cross-scan map after
individual drift correction

6. Final map

The **photProject** task performs a simple co-addition of the images using the drizzle method (Fruchter and Hook, 2002, PASP, 114, 144). The key parameters are the output pixel size and the drop size (**pixfrac**). **A small pixfrac value can help to reduce the correlated noise due to the projection.**

```
Console x
HIPE> finalMap, mi = photProject(mergedScans, outputpixelsize = outputpixelsize, pixfrac =
pixfrac, caltree = calTree, useMasterMaskForWcs = False)
```

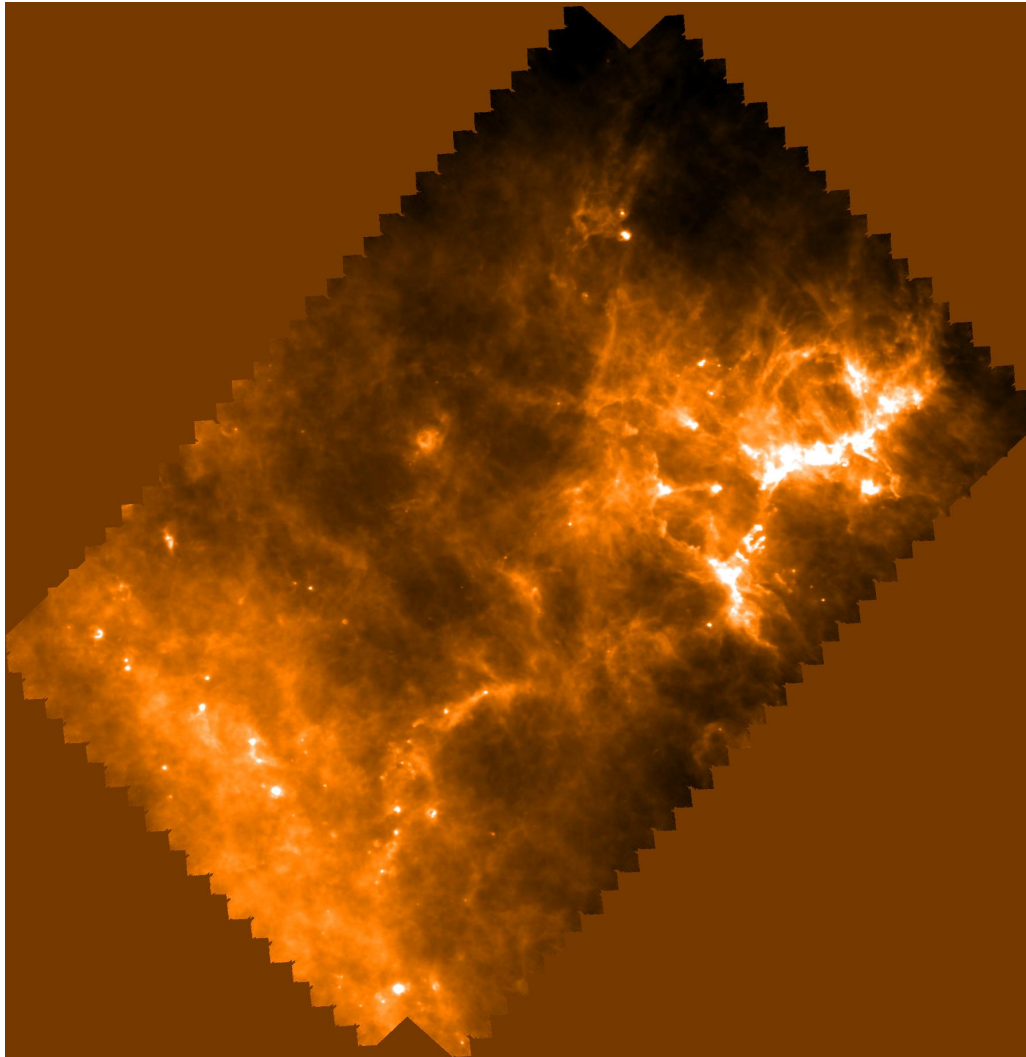
NOTE: *photProject* is the same task used by the High-Pass Filter pipeline to generate the final map

Default Values:

- ❖ pixfrac = 0.1
- ❖ parallel observation or red band: outputPixelSize = 3.2"
- ❖ blue band in PACS scan mode: outputPixelSize = 1.6"



.....and here it is



PACS-402



Useful Notes

- Jscanam accepts Level 1 data directly from the Herschel Science Archive (HSA). No conversion is needed, such as in IDL Scanamorphos
- The Jscanam ipipe script requires virtually no parameter tweaking and can be run as is for the vast majority of science cases. In particular, the **galactic** parameter called by many tasks can be left set to **True** (default) even when processing extra-galactic data sets
- The current Jscanam ipipe script (HIPE 12) can process only 1 pair of OBSIDs at a time. Multiple pairs can be process by joining the scans as described in the script itself (→ see warning message). In HIPE 13, a new JScanam ipipe script will become available which will allow the processing of multiple pairs starting from Level 1
- Jscanam does not allow processing of turnaround scans. If these scans are needed, we recommend using other mappers (i.e. MADmap or Unimap)



Documentation

For more information on Jscanam, check:

- Section 3.2.3 of the PACS Data Reduction Guide:

http://herschel.esac.esa.int/twiki/pub/Public/PacsCalibrationWeb/pacs_phot_Hipe12.pdf



Thank you !