MADmap User's Guide for PACS

Author: Babar Ali
Version: alpha
Document ID: PICC-NHSC-026

History

Version	Date	Author	Change Record
alpha	10-Apr-2010	Babar Ali	First write up.

Table of contents

1 Introduction

The Microwave Anisotropy Dataset mapper (MADmap) is an optimal map-making algorithm, which is designed to remove the uncorrelated one-over-frequency[1] (1/f) noise from bolometer time ordered data. See RD3 for details on how the MADmap algorithm solves for the optimal signal. Figure 1 illustrates the use of MADmap on one particular dataset for PACS. The removal of 1/f noise creates final mosaics without any so called "banding" or "striping" effects.

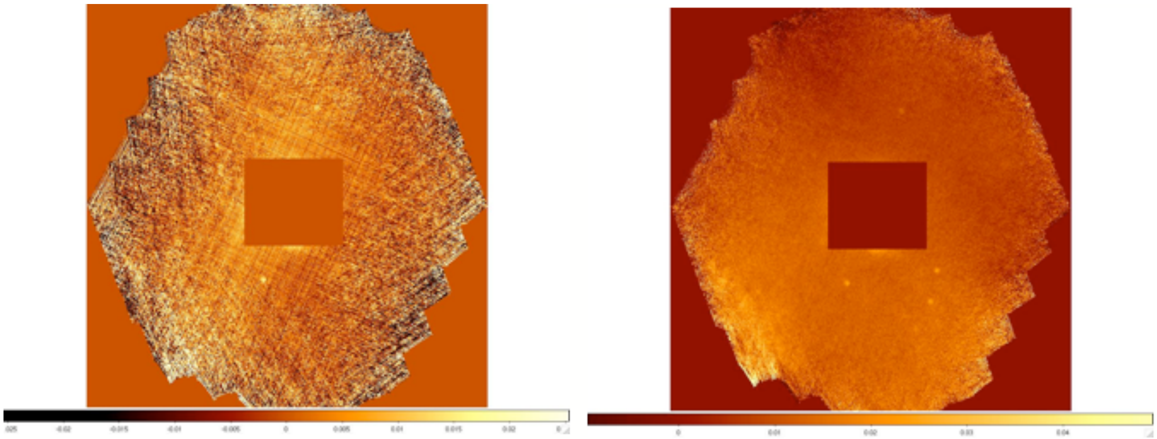


Figure 1 The function of MADmap is to remove the effect of 1/f noise. Left: image created without MADmap processing. Right: The same image after MADmap processing. The central object has been masked out.

For Herschel data processing, the original C-language version of the algorithm has been translated to java. Additional interfaces are in place to allow PACS (and SPIRE) data to be processed by MADmap. This implementation also requires that the noise properties of the detectors are determined apriori. These are passed to MADmap as PACS calibration files and referred to as the INVNTT files or "noise filters".

2 Level 0 to Level 1 processing.

The time streams must be instrument artifacts free and calibrated before MADmap can be used to create the final mosaic. This is a two-part process. The first part is the PACS level 0 (raw data) to level 1 (cleaned and calibrated images) processing. This is discussed here. The second part is MADmap pre-processing, which is discussed in the next section.

---

[1] 1/f noise is defined as noise whose intensity is inversely proportional to its Fourier frequency to some power.

For most users standard Level 0 to Level 1 processing is normally sufficient. However, the method used for deglitching the data may have a significant and adverse impact on MADmap processing. For MADmap, we recommend and prefer the IInd level deglitching option. This option is not part of the automated ("standard") pipelines. The alternative, the "standard" wavelet based "MMTdeglitcher" does not perform optimally when it is allowed to interpolate the masked values. If the MMTdeglitcher is used, we recommend selecting the 'maskOnly' option to prevent replacing masked values with interpolated ones.

The HIPE task 'MADmap_procL0' is the recommended task for processing raw PACS data for the MADmap pipeline.

3 MADmap Pre-processing.

The point of using MADmap is to account for signal drift due to 1/f noise while preserving emission at all spatial scales in the final mosaic. This is fundamentally different from the high-pass filter reduction, which subtracts the signal at scales larger than the size of the high-pass filter window. However, the MADmap algorithm, indeed most optimal map makers, assume and expect that the noise in the time streams is entirely due to the so-called 1/f variation of said detectors. The PACS bolometers show correlated drifts in the signal that must be mitigated before MADmap can be used. The MADmap algorithm assumes that the noise is not correlated and will incorrectly interpret the any systematic non-1/f-like drifts as real signal. Additionally, the PACS bolometers have pixel-to-pixel electronic offsets in signal values. These offsets must also be homogenized to a single base level for all pixels.
The mitigation of all of the above effects is referred to as MADmap preprocessing. In all, there are 4 different types of corrections. We discuss each step below.

WARNING: The MADmap preprocessing critically determines the quality of the final maps. Care must be taken to ensure that each step is optimally applied to achieve the best possible reduction. This may require repeating step(s) after interactively examining the results. Further, not all steps may be necessary. This is also discussed below.

3.1 Pixel-to-pixel offset correction.

This is the most dominant effect seen in all PACS readouts. For most single channel bolometers the offset is electronically set to approximately 0. The PACS bolometers are, however, multiplexed and only the mean signal level for individual modules or array can be set to 0, leading to the observed variations in the pixel-to-pixel signal level. This is purely an electronic and design effect.

Mitigation of this effect entails subtracting an estimate of what the zero level should be per pixel from each of the readouts of the pixel. There are two ways to estimate the zero level.

(1) Use a calibration zero-level image. The following snippet shows one example, for the blue filter, of how to access the zero-level image In HIPE.

```
HIPE> calTree = getCalTree()
HIPE> zimage = calTree.photometer.corrZeroLevel["blue"].data
```

Where, "blue" in the square brackets should be replaced with the appropriate filter for the observations.

The resulting variable 'zimage' is a 2D floating-point array, and this should be subtracted from each of the signal readout. The units for the zero-level image are volts. Therefore, the readouts must also be in volts. **Figure 2** shows the calibration zero-level image for the blue filter.

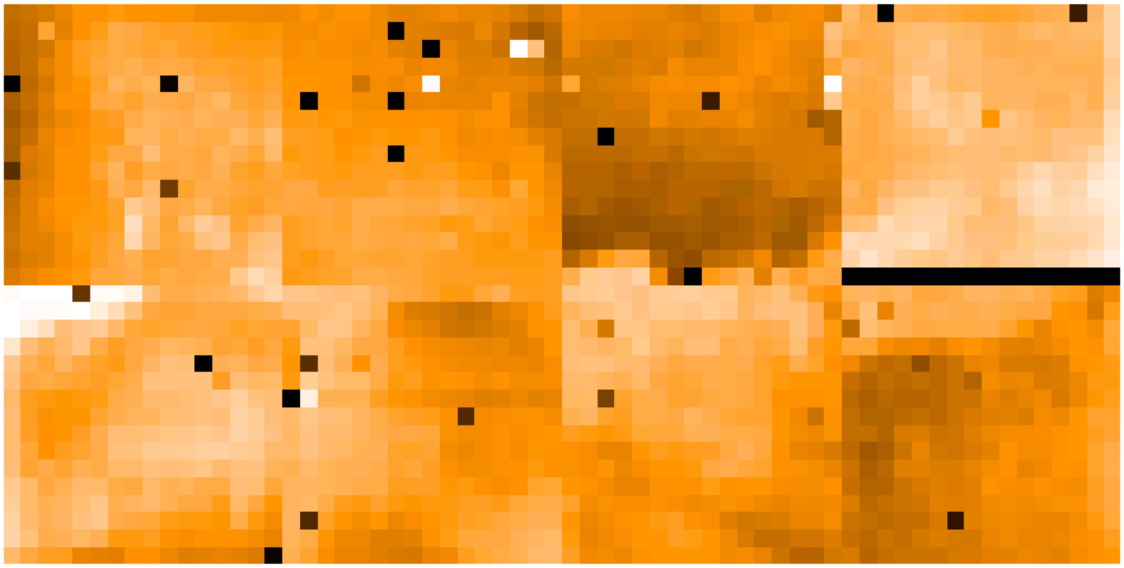


Figure 2 The zero-level calibration image showing the pixel-to-pixel electronic offset.

(2) Use the median of the entire pixel history.

This method works in any units (digital readout units or volts). The idea is to compute the median of the entire history of signal values per pixel and subtract this median from each of the pixel readouts. The task photOffsetCorr() applies this correction in HIPE. For example,

HIPE> frames=photOffsetCorr(frames)

Figure 3 shows the resulting readout from **Figure 2** after pixel-to-pixel offset correction using the photOffsetCorr() task in HIPE.

Tip: The photOffsetCorr task uses the on-target status flag (OTF) to determine which readouts are to be used to estimate the offset values. This flag can be manipulated to specify, among other possibilities, which part of the sky (bounded by right ascension and declination values) is suitable for estimating the offset values. Simply set the OTF flag to false. Then, set the OTF flag as true for all readouts that are within the boundaries of the sky region.

3.2 Module-to-module drift correction.

Figure 3 also shows that the blue focal plane readout, after the offset correction, has a systematically different signal level in the two modules appearing on the bottom left of the Figure than the remaining modules.

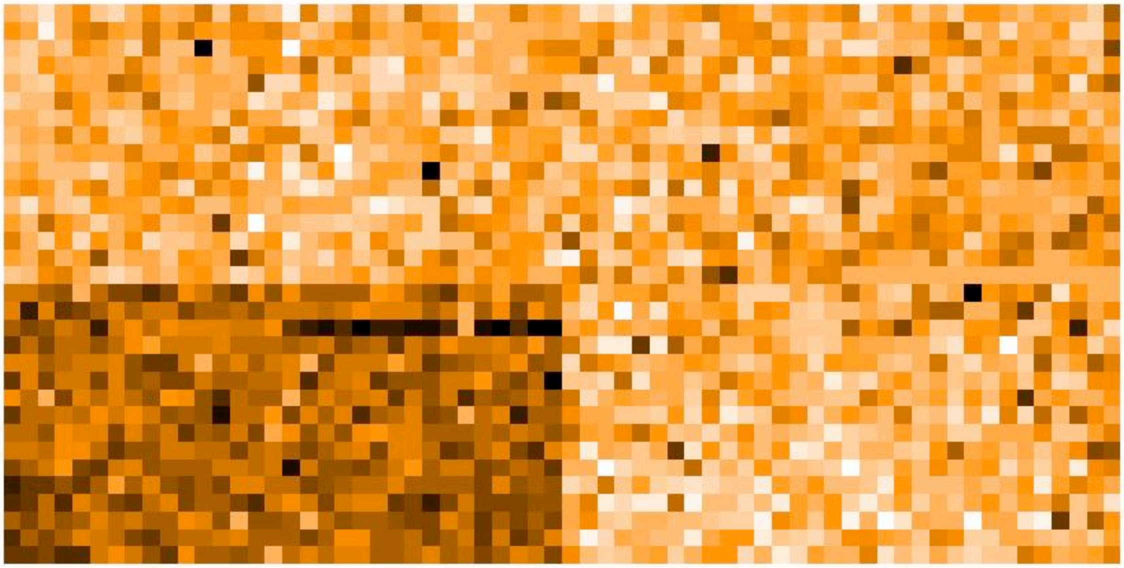


Figure 3 The raw signal frame after the offset (pixel-to-pixel) image removal step. While the pixel-to-pixel variation is mitigated, the result shows two modules are systematically at a different signal level than the rest.

If we had used a different offset correction, different pairs of modules may appear errant, but always in pairs (1&2, 3&4, 5&6, or 7&8). Generally, the pair 1&2 is always systematically off. Mitigating this systematic difference in the signal level between individual modules is what we refer to as 'module-to-module' drift correction.

Figure 4 shows why drift is an apt description for this correction. In this Figure, we show the result from a more quantitative investigation, in which the median level of the module is subtracted from a reference module (in this case #5) and plotted against the readout index (or time). The deviant signal "spikes" in the data are due to actual variation in the sky signal. The apparent "break" in the trend around index 14800 is the difference between scan and cross-scan readouts that have been merged in this Figure. Individual modules are color-coded as labeled below the Figure.

Figure 4 shows that the relative median levels of the modules are: (i) not the same, and (ii) change systematically as a function of the readout (or time). As is the case for the pixel-to-pixel variation, the MADmap code will also not handle such correlated module-to-module variation.

At present, the recommended mitigation is as follows: (i) select only data with median level between +/- 50 counts, or +/-0.1 volts, depending on the signal units. This will more or less ignore the real astrophysical variations in the signal. This is a practical, but not necessarily the optimal way to de-trend module-to-module variations. (ii) Fit a straight line to the data as a function of the reset index. (iii) Subtract the fit from all pixels of the module.

In HIPE, the task 'photModuleDriftCorrection()' applies the steps listed above automatically. It is called, for examples, as follows:

```
HIPE> frames = photModuleDriftCorrection(frames)
```

See the documentation within the photModuleDriftCorrection task for information on optional parameters.

Figure 5 shows the resulting linear fits to the module drifts. Some fits were affected by the larger variations caused by the real sky brightness variation over the scanned region. Even for these, the resulting discrepancy is small. There is no evidence that a higher order model is needed.

Figure 6 shows the resulting readout (same index as the one in Figure 1) after the best-fit signal drift is subtracted from each of the module's pixels.

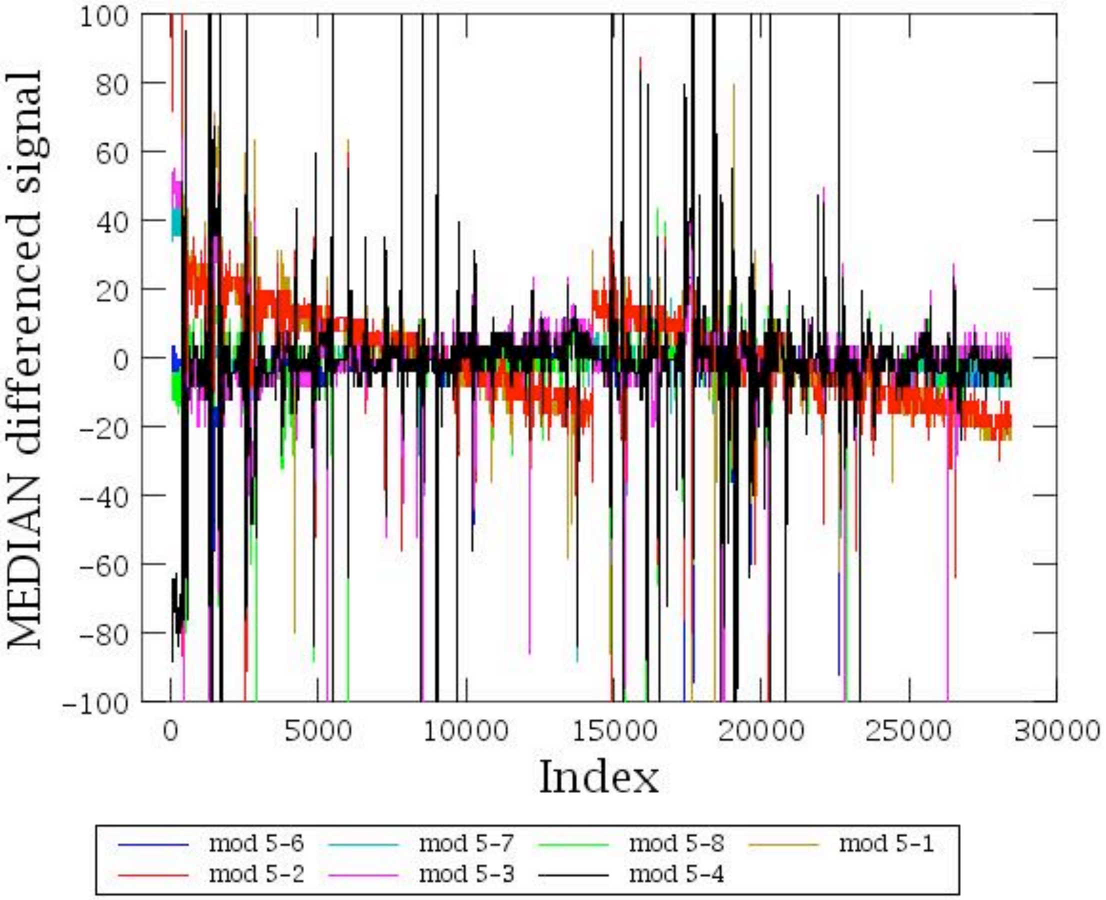
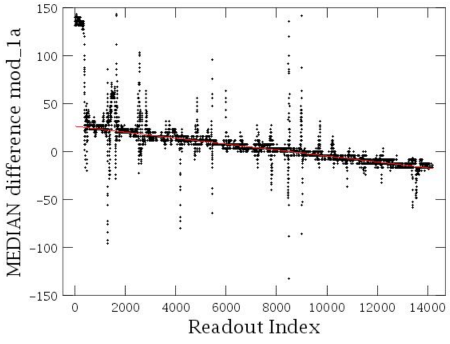
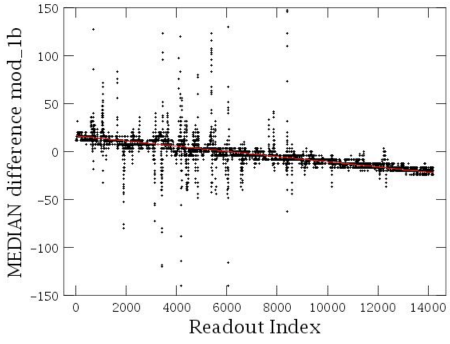
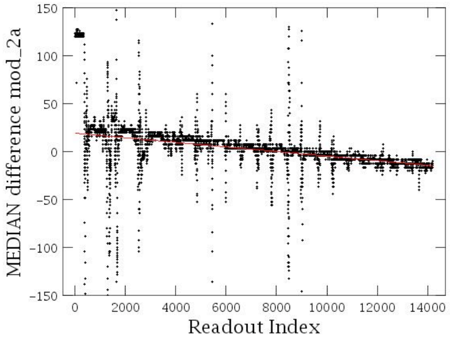
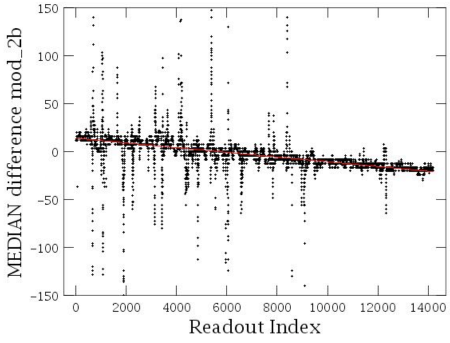


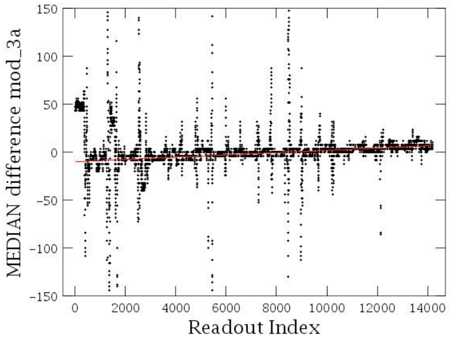
Figure 4 The systematic module-to-module drift. The difference is shown in the median level of the module minus the reference module as a function of readout index. The most discrepant modules are 1&2 (shown in reddish tints) but all appear to show some drift in the median level of the frame.

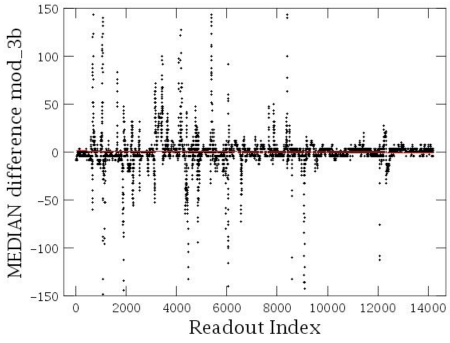

MEDIAN difference mod_1a
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index

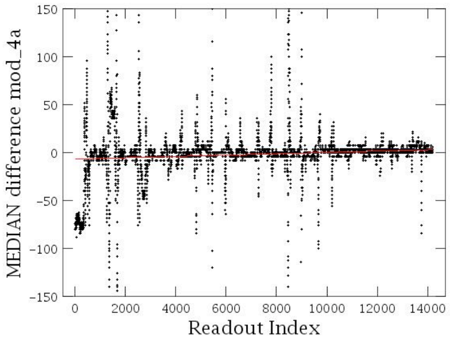

MEDIAN difference mod_1b
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index

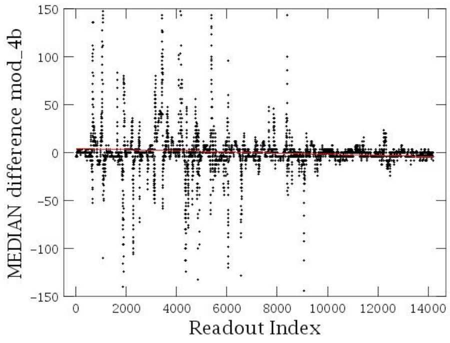

MEDIAN difference mod_2a
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index


MEDIAN difference mod_2b
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index


MEDIAN difference mod_3a
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index


MEDIAN difference mod_3b
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index


MEDIAN difference mod_4a
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index


MEDIAN difference mod_4b
150
100
50
0
−50
−100
−150
0
2000
4000
6000
8000
10000
12000
14000
Readout Index

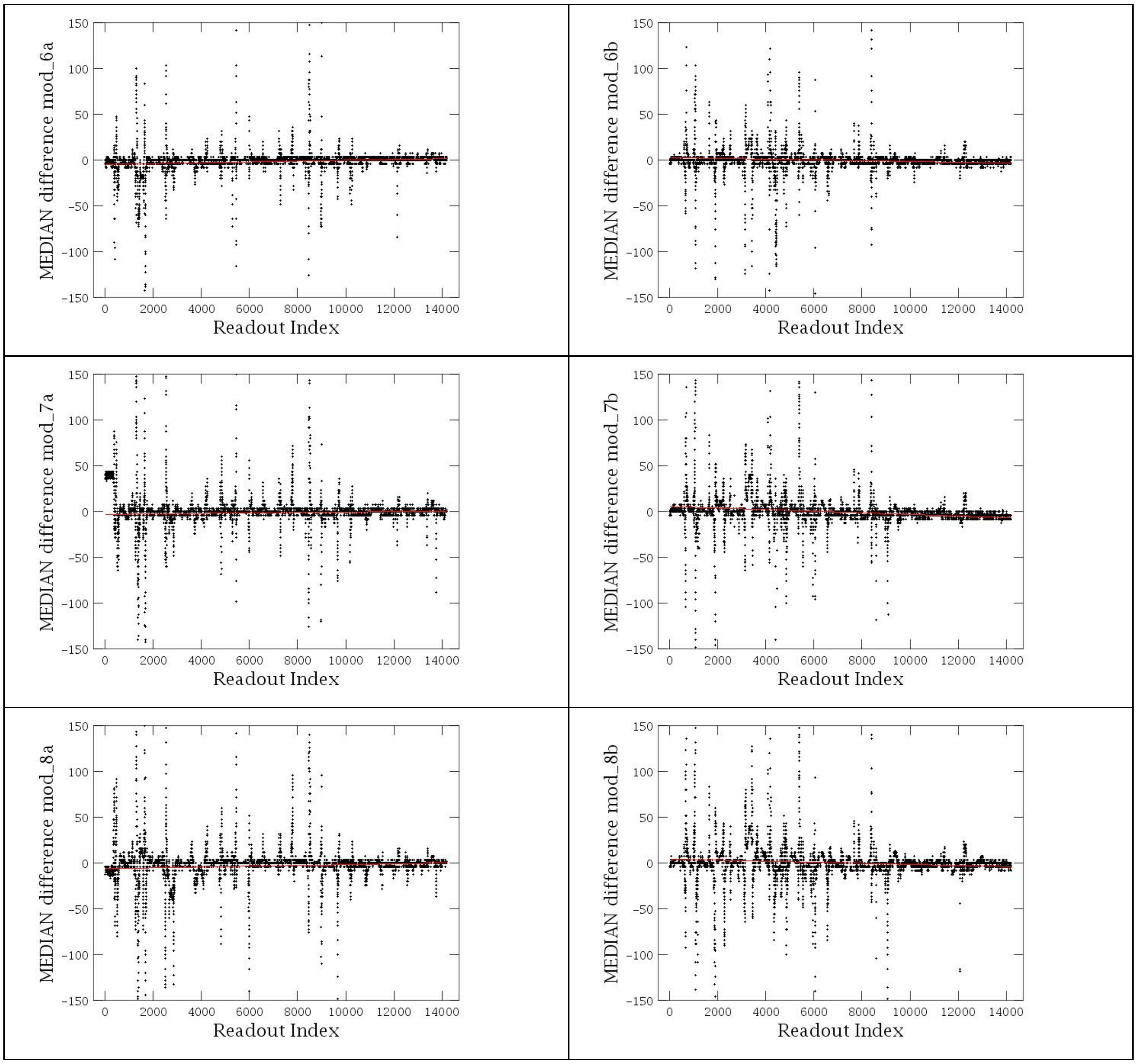


Figure 5 The best-fit linear model to the module-to-module drifts (red line). The left column shows the fits to the scan readouts, the right column the same to the cross-scan readouts.

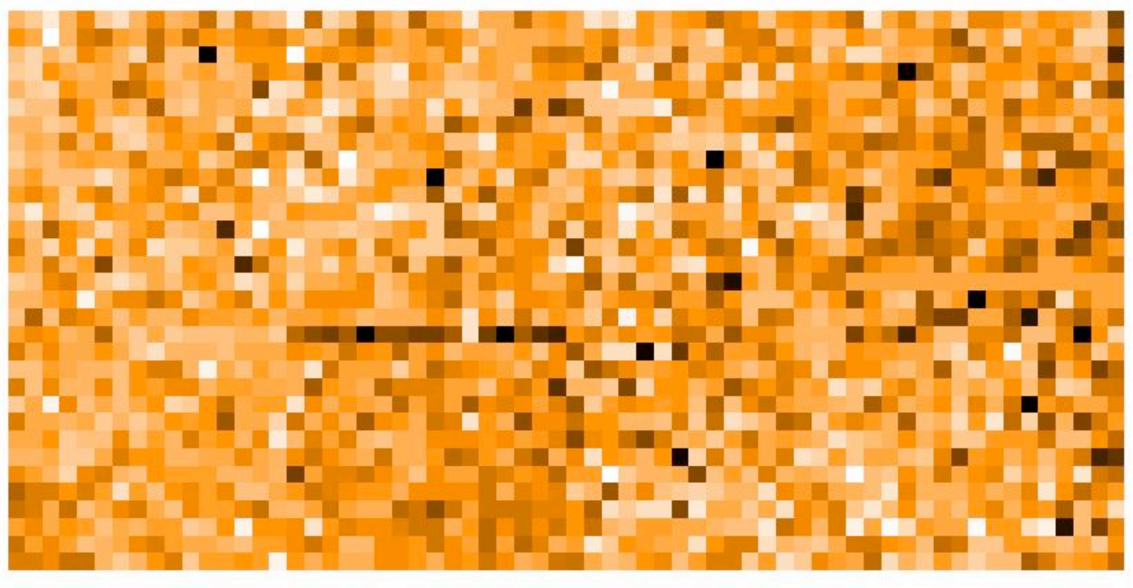


Figure 6 The frame from Figure 1 after module-to-module correction as described in the text.

The module-to-module correction is not required for the red channel.
After the module-to-module drift correction, the data are such that systematic differences in the median level of the frame are not significant.

3.3 Global signal drifts correction.

Since signal for individual modules drifts in relation to the reference module, most assuredly, we expect the reference module itself to drift as well. The effect of such a drift produces detector readouts that are systematically offset in signal from the beginning to the end of the scan in a monotonic way. Once again, this is a global correlated trend of the entire array and, must, therefore, be removed prior to MADmap.

Establishing a systematic drift in the signal of the reference module is much more difficult as no "constant" or "relative" comparison data are available. Figure 7 shows the median signal of the entire array as a function of the readout index (time). The reader may be able to convince his/herself that a trend exists despite the sources coming on and off the array. The larger variations are from actual source/sky signal, which happen to dominate the readouts for this observation. The trend is more clearly visible in Figure 8 because the sky/source signal is much weaker compared to the background.

For data that are dominated by the background (as in Figure 8) the trend is relatively easy to model as source contamination is negligible. However, a more generic approach is needed that is also able to account for data that are similar to the one shown in Figure 7. To do this, we create 1000 readout wide bins and assumed that the minimum values in these bins will correspond to the few actual "blank"/background measurements. The idea is that over these 100 readouts, at some point, the scan pattern observed a "source-free" part of the sky.

Figure 9 illustrates these minimum values plotted as a function of the readout index where they are found. The dots in Figure 9 are the actual minimum median values in each of the 1000 readout bins. The trend is now much easier to model. Figure 9 shows a 2nd order polynomial model to the data. The best-fit model (polynomial in this case) is then assumed to described the correlated global drift and is subtracted from the each readout.

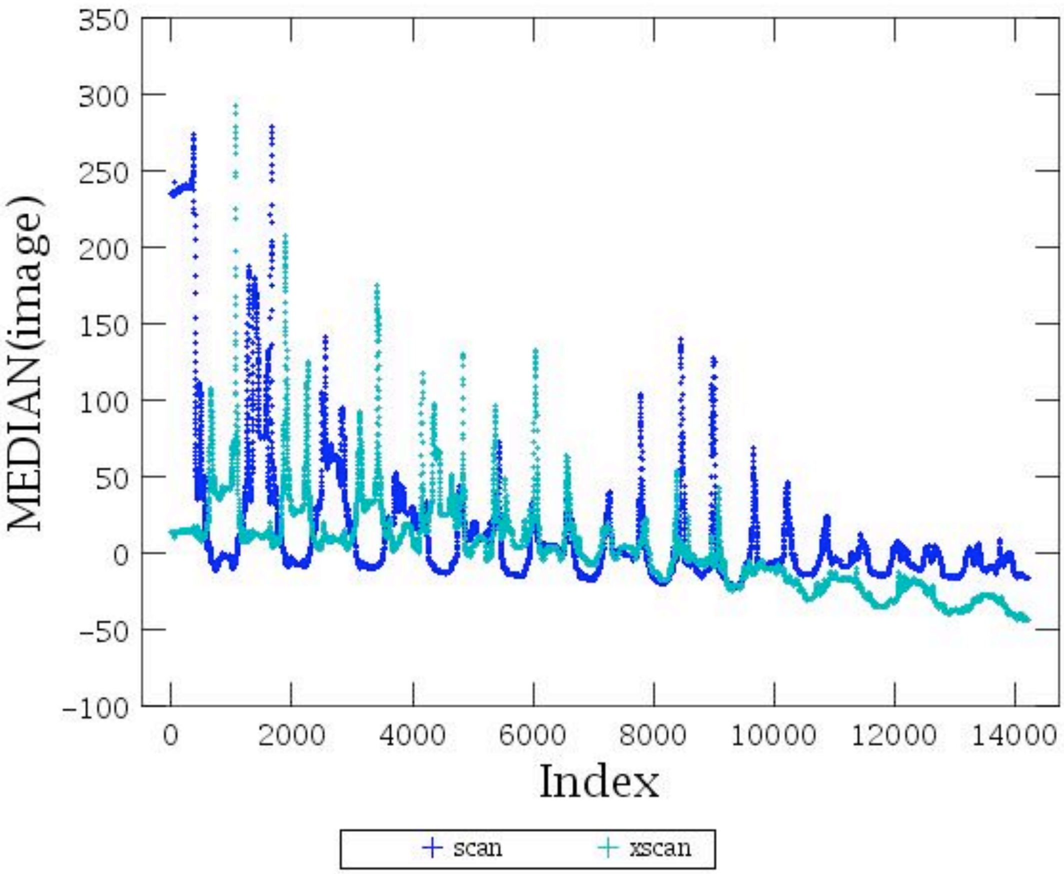


Figure 7 The global median of the image as a function of the readout for the scan and xscan data. Is there a trend downwards in the signal?

The task photGlobalDriftCorrection() applies this technique automatically. In HIPE the calling sequence is, for example:

```
HIPE> frames = corrGlobalDriftCorrection(frames)
```

See documentation within the corrGlobalDriftCorrection task for the API.

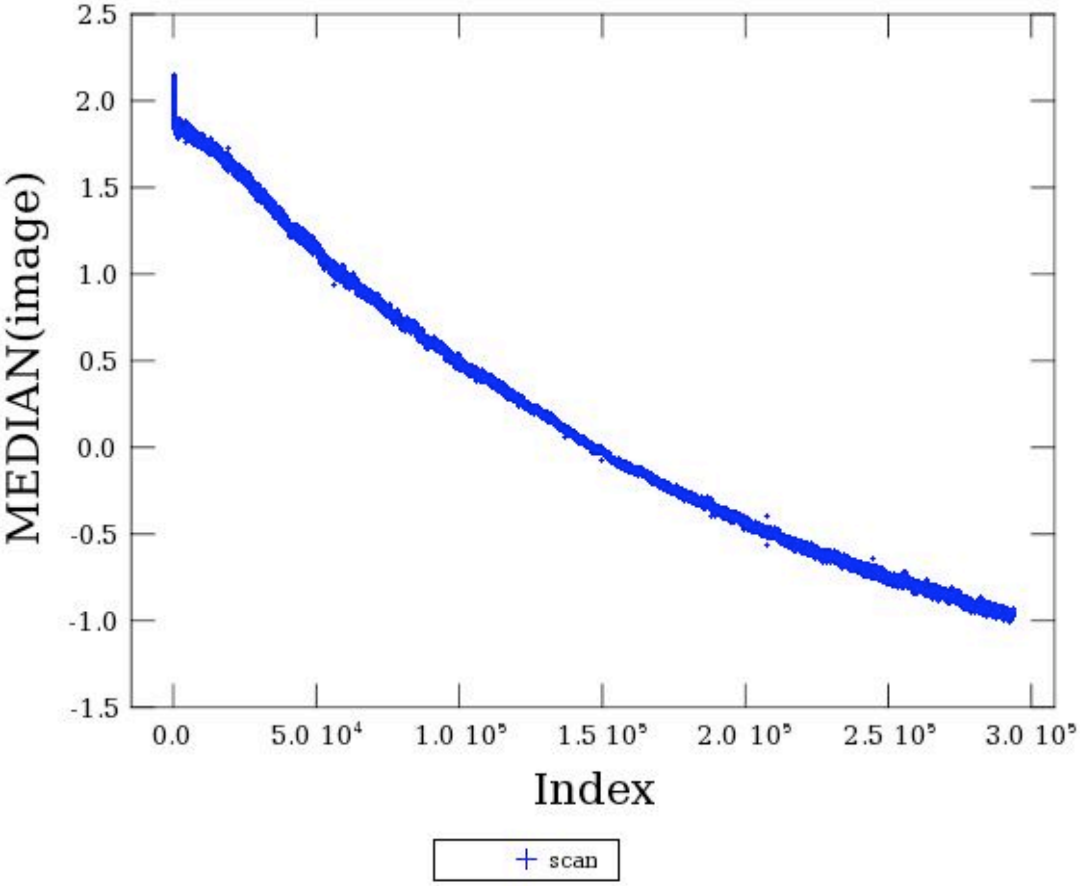


Figure 8 The median value of the entire PACS array as a function of readout index (time). The monotonic trend seen here is due to the correlated drift in the signal of the entire array. The 1/f noise variations are much less significant compared to the overall signal drift.

TIP: To reduce (or prevent) frequent reprocessing of the same steps, it is recommended that you save your data prior to applying global signal drift corrections. This way, if you wish to examine the relative merits of various correction options, the processing up to this stage does not have to be repeated.

3.4 Global drift correction models

The global signal mitigation is much more important than the module-to-module signal drift, as this step affects the quality of final maps much more significantly. In this section, we discuss what options exist for modeling the global drifts.

Option 1: This is the approach already discussed above. The fit to the baseline is a polynomial. For most data a 2nd order polynomial provides an accurate fit. For a significant fraction of the data, particularly those with less than few thousand readouts, a linear model has sufficient accuracy. This approach is useable for all data.

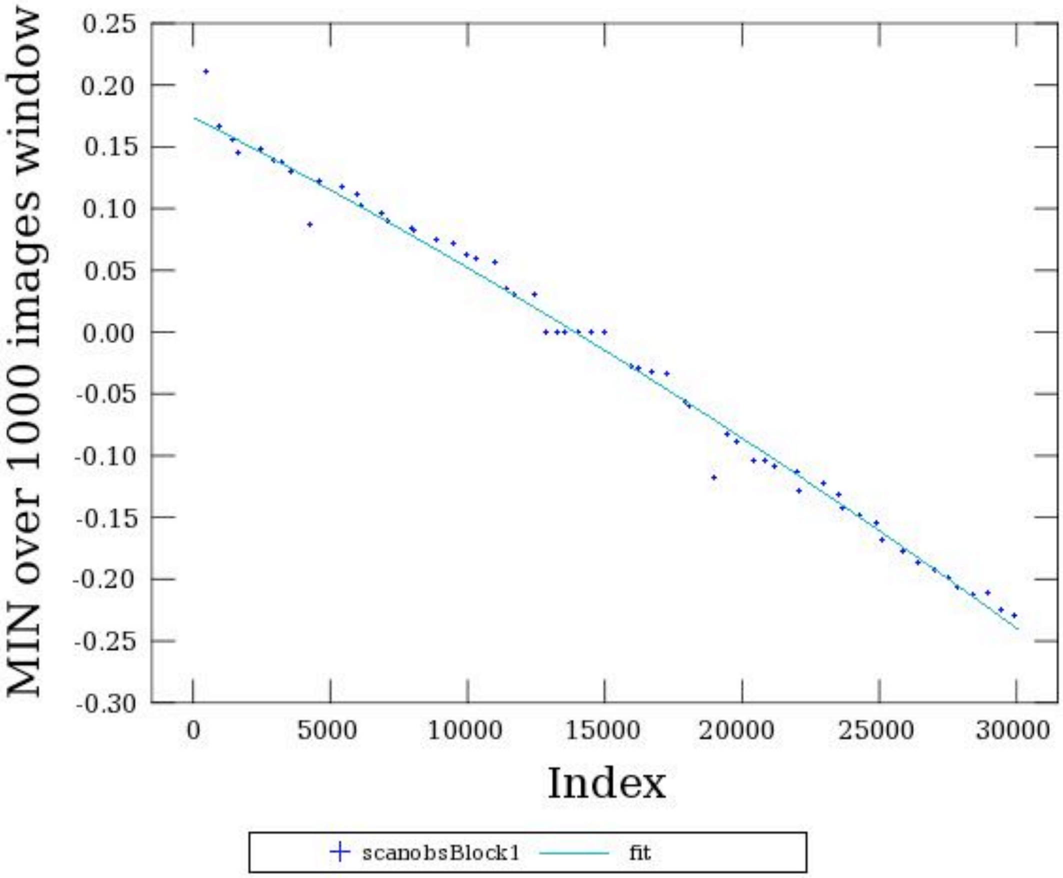


Figure 9 The minimum value of the 1000 readout bin as a function of its corresponding readout index number (see text for details). The dots show the actual minimum of the median. The line shows a modeled fit to the data.

Option 2: For data that are similar to the one shown in Figure 8, i.e., the signal is dominated by the background and not source emission, another option is to simply subtract the median value of the entire array from each pixel. This approach will fail when the sky emission is significant, even for a few readouts.

A significant disadvantage for both option 1 and 2 is that individual pixels are known to show small variations in their signal drift that is different from the global signal drift of the array. This leads to noisier maps than would be possible if drifts from individual pixels can be modeled. However, the magnitude of the additional noise component is relatively small. There is, unfortunately, no real cure for this for option 1. However, in some cases, option 2 can be modified as described below in option 3.

Option 3: Again for background dominated data, individual pixel time lines can be fit with a polynomial and subtracted. This approach has the advantage that individual pixel time lines are fit and subtracted. However, it has the disadvantage that for individual pixels the trend is less obvious and is also affected by the 1/f noise.

3.5 Which option to use?

For observations that are ~few X 10^3 readouts or less in size, we do not recommend any global signal drift correction. Over such short readouts, the detectors have not sufficiently drifted to cause a significant variation and will not adversely affect the final maps. The user can judge the need for this correction by examining the median values of the array for their data set. However, having stated that, we also note that observations taken at the beginning of the PACS observation day (OD) are much more susceptible to drifting signal than those taken at the end of the OD.

For observations longer than ~few X 10^3 readouts, we recommend option 1. The default setting should apply to most observations. For the best-possible results, we recommend interactive optimization described below. This is also the setting recommended for the default automatic pipeline.

If the observations are dominated by background (not source) emission, option 2 and option 3 provide more accurate rendering of the final maps. However, care must be taken to first establish that said assumption is valid. This approach is not recommended for automated processing.

4 Optimizing MADmap pre-processing.

As mentioned earlier, the quality of the final map is strongly dependent on the quality of the preprocessing steps. The default settings now programmed for the various HIPE tasks/modules already introduced produce reasonable maps. However, baseline removal is a tricky art and the user can further optimize their maps by following the following suggested interactive steps.

4.1 Overriding defaults in option 1.

Option 1 is implemented as model 1 in photGlobalDriftCorrection task (see the Appendix for the API). Both the bin size over which the minimum is calculated, as well as the order of the polynomial, are controlled by the user. Smaller bins more accurately model the drift provided that source emission does not creep into individual bins. And, the best-fit model may be a higher order polynomial. The users must decide this for themselves by investigating the proper values for both parameters and examining the final fits.

4.2 Segmenting time-streams.

For observations longer than ~2 hours, more global inflections become visible in the data. Figure 10 illustrates this behavior. The beginning section of the observation is not well-fit with the 2nd order polynomial model used. In fact, while not shown, we find no satisfactory fit for the data with any order polynomial. The best solution in this case came from segmenting the data into 30,000 readout groups. An example of the fit to one individual group of 30,000 is shown in Figure 9. This approach produces much more sensitive maps than a forced fit to all data. Figure 11 shows a comparison of the resulting maps with and without segmented fitting. The "structure" visible in the map without segmented fitting disappears when the drift correction is applied to individual segments. We attribute the difference to inaccuracies in modeling the signal drift without segmenting.

This approach is not recommended for smaller datasets because: (i) not enough data are available for grouping, and (ii) when the segments become smaller than the scan length, actual spatial structure will be removed, hence, negating the use of MADmap.

4.2.1 Optimal segmenting.

We recommend that the minimum segment should be on the order of a few scan legs to ensure that spatial structure of the size of the map length are preserved, and enough data are present for proper binning. In reality, a combination of segment and bin size is best derived empirically be interactively examining the fits.

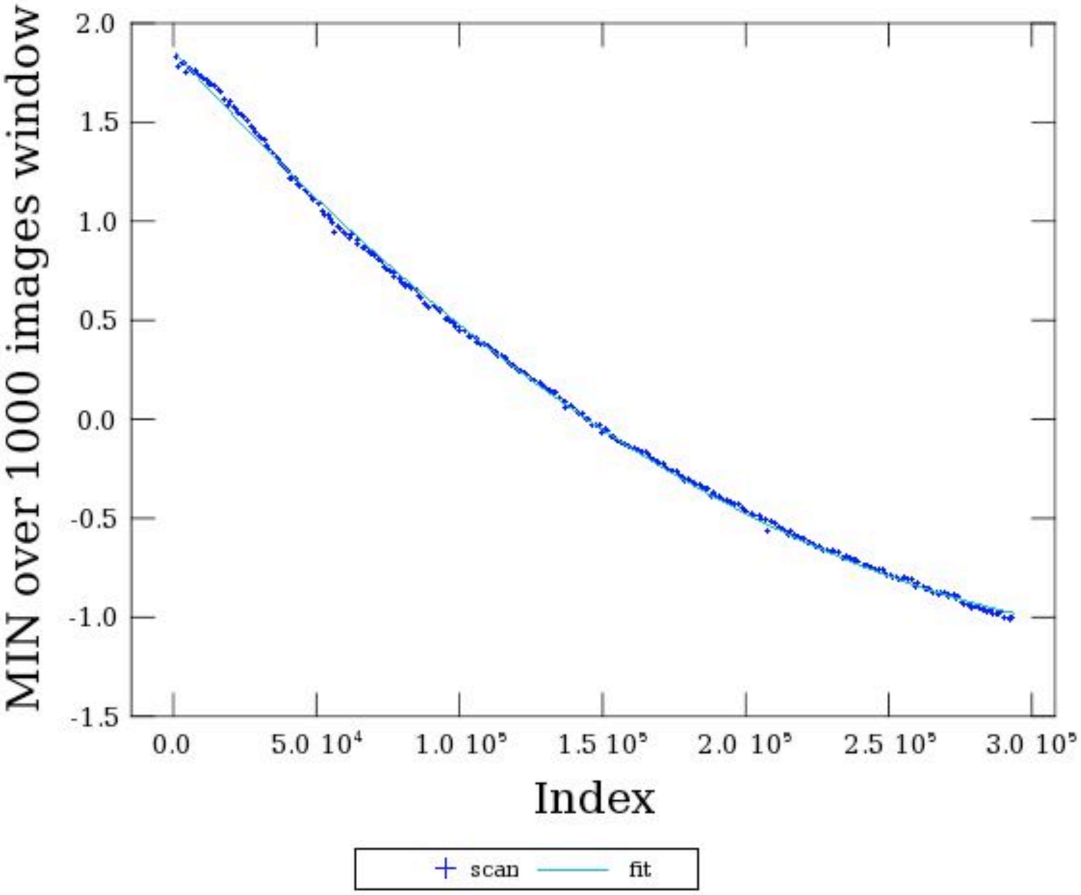


Figure 10 Option 1 and a 2nd order fit applied to an 8 hour duration data. the best fit is reasonable towards the end of the observations but fails to match the inflections at the beginning of the observation.

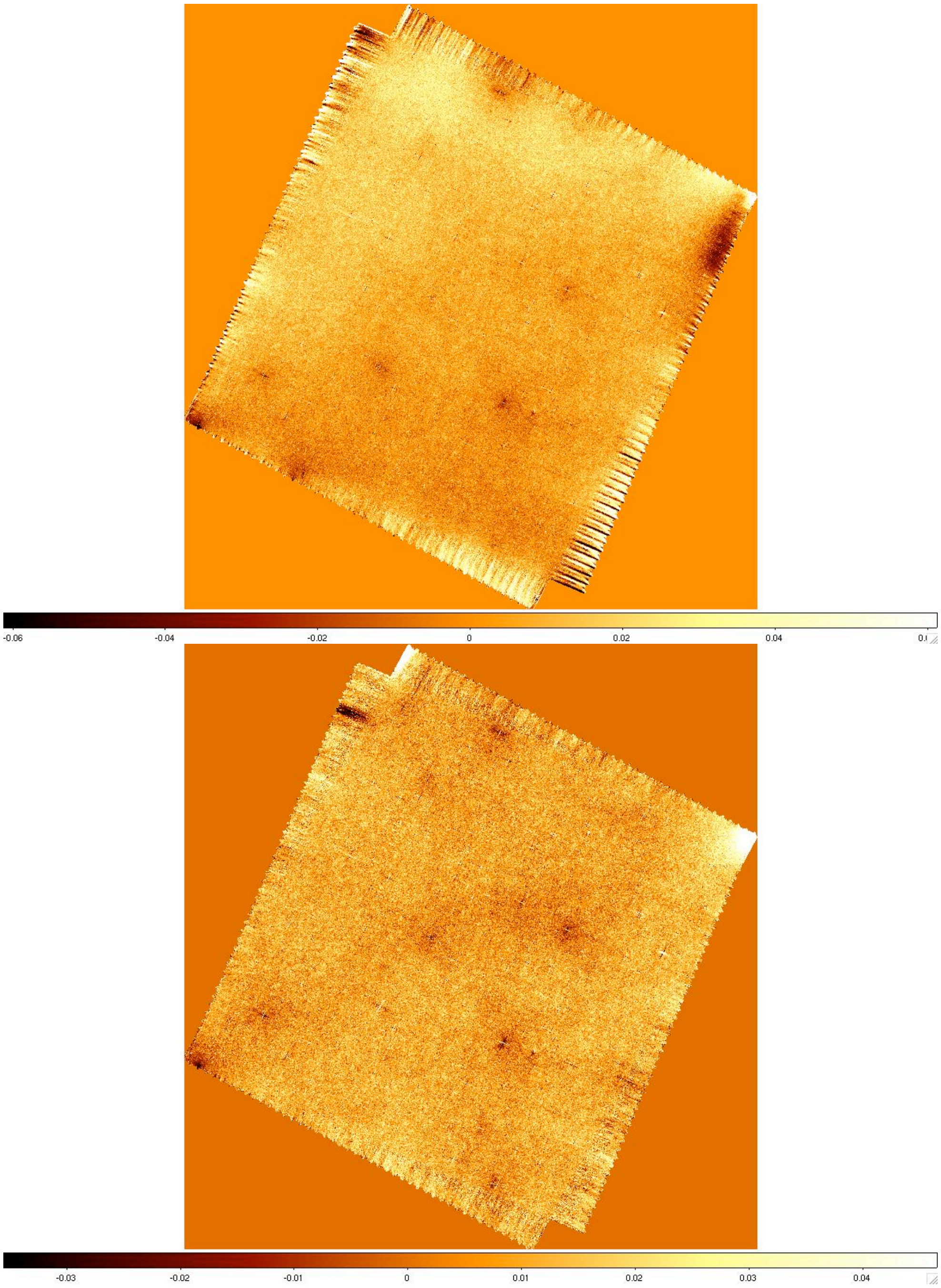


Figure 11 Comparison of final maps from default option 1 setting (top) and segmented option 1 fitting (bottom). The global signal drifts are much better corrected when large data sets are fit in smaller segments.

4.3 “long-pass” filtering.

Another way to combat the global drifts is to preprocess with a so-called “long-pass” filter. This is a variation of the high-pass filter except that filter window sizes of 500-3000 readouts are used. As for the high-pass filter, the primary disadvantage of this approach is the removal of all spatial structures of sizes larger than the long-pass filter width. The primary advantage is gained in more accurate removal of the global signal drift. However, the disadvantage is such that for data similar to the one shown in Figure 7, they simply do not work and leave significant filtering artifacts in the final maps. If long-pass filter is used, then the photGlobalDriftCorrection() task should be omitted from MADmap processing.

4.4 Other global drift correctors.

The above methods are already available in HIPE. However, it should be noted that users are free to choose any other established or customized drift correction model that they deem appropriate, provided it can be programmed in HIPE. Examples include: interpolated or tabulated drift correction per readout based on the minimum median values. In this case, the actual minimum medians (the dots in Figure 9, for example) are used to create an interpolated lookup table for each readout.

5 1/f noise removal and MADmap.

After pre-processing, in theory the only drift left in the signal should be due to the 1/f noise. Figure 12 shows the power spectrum of the data cube (averaged for all pixels) after the drifts are accounted for. The *a priori* selected noise correlation matrix for PACS is estimated from Fourier power spectrum of the noise.

The current MADmap implementation requires the following data to be present in the frames object:
Camera
(RA, Dec.) cubes
BAND
OnTarget flag
These data are generated from Level 0 to Level 1 processing, or in the Level 0 product generation. MADmap will not work if any of the above dataset or status keywords are missing.

Camera. Should start with ‘Blue’ or ‘Red’. To check, use the following command in HIPE:

```
HIPE> print frames.getMeta()["camName"]
{description="Name of the Camera", string="Blue Photometer"}
```

The (RA, Dec.) cubes are the 3-dimensional double precision cubes of Right Ascension and Declination values generated during level 0-1 processing. In HIPE:

```
HIPE> print frames[“Ra”].data.class
```

If an error occurs (provided no typos are present) then the (Ra, Dec) cube simply hasn’t been generated.

The BAND status keyword must have one of ‘BS’, ‘BL’, or ‘R’ values. If BAND does not have one of these values, MADmap will not work.

```
HIPE> print frames[“Status”][“BAND”].data[0:10]
```

OnTarget status keyword. This in a Boolean flag under status and must have the value ‘true’ or ‘1’ for all valid sky pixels for MADmap to work. E.g.:

```
HIPE> print frames[“Status”][“OnTarget”].data[0:10]
```

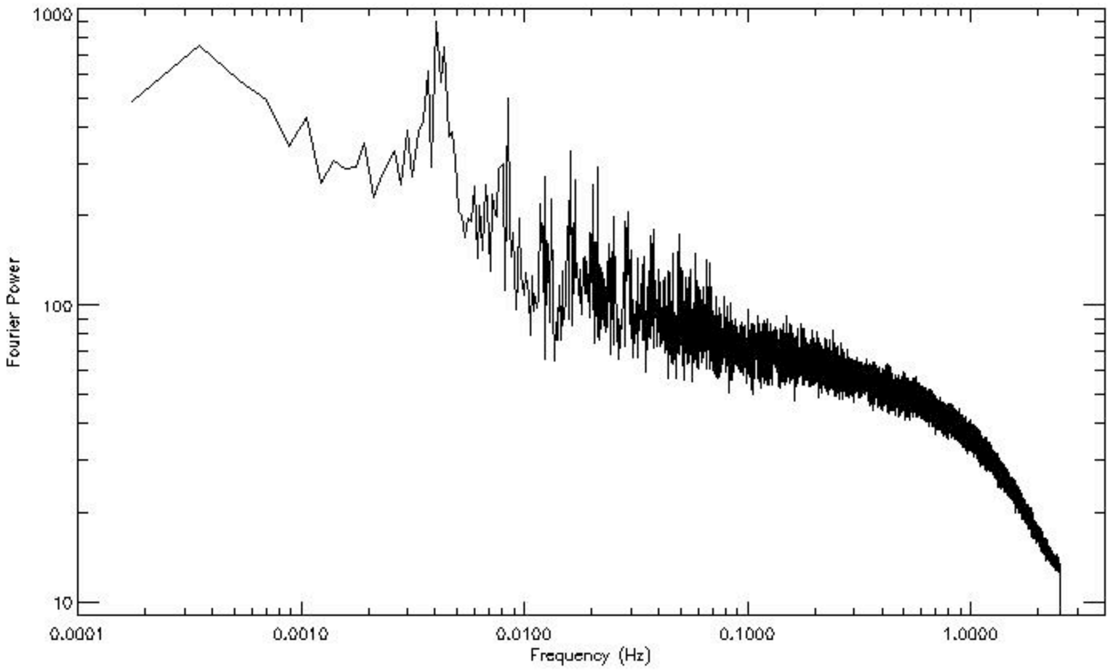


Figure 12 The power spectrum of the full data stream after the drift removals (averaged for all pixels). Some structure is expected due to the astrophysical sources and from the un-removed glitches.

The data are now ready for MADmap. For API description on how to run MADmap, please consult RD4.

6 Flux calibration and sensitivity.

The accuracy of this map compared to high-pass filtered maps is discussed in a companion report RD2. In summary, MADmap produced maps have absolute flux levels consistent with those produced with the alternative high-pass filter and projection algorithms to within 5%. The sensitivity, as measured by the rms in the sky value and photometry of individual sources, is ~40% worse in MADmap produced maps than in high-pass filtered projections.

We present one additional comparison between the high-pass filtered mosaics and MADmap. This one is for the ATLAS key program. The high-pass filtered data are from photometry of sources for which *a priori* locations are known from SPIRE data. We use the same locations in the MADmap mosaic to measure the photometry. Figure 13 shows the comparison. The findings in this field confirm the conclusions of RD2.

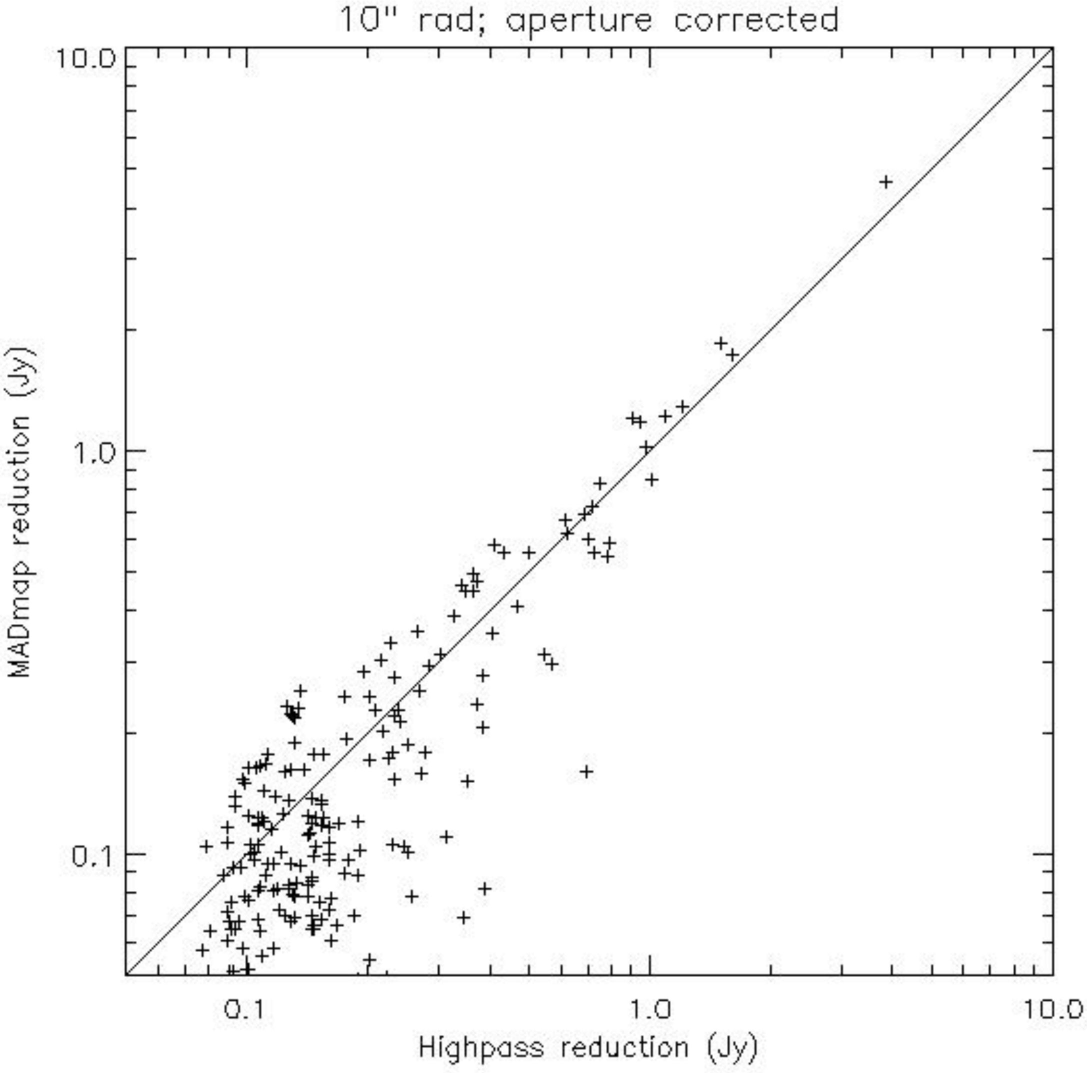


Figure 13 Comparison of photometry values between MADmap and highpass filter reduction for the same field. The results here support the conclusions of RD2.

7 Open issues and known limitations.

The following items are known limitations of MADmap processing:

7.1 Point source artifact.

TBW.

7.2 Computing requirements.

TBW

In addtion, the following is a list of open issues:

- What is the origin of module-to-module offset and drift?
- Is the global signal drift repeatable and correlated with time since last recycling?
- Is there a Housekeeping parameter that mimics the global signal drifts?

8 Trouble shooting

This section captures various flavors of what might go awry with MADmap processing.

8.1 Glitch in the readout electronics.

If a cosmic ray (or a charged particle) impacts the readout electronics, the result may be a significant change in the drift correction for the array (or the module) as a whole. Figure 14 illustrates this. Figure 15 shows the smoking gun that caused the global drift to change.

The only possible remedy is to segment the data before and after the glitch even and fit the global drift separately.

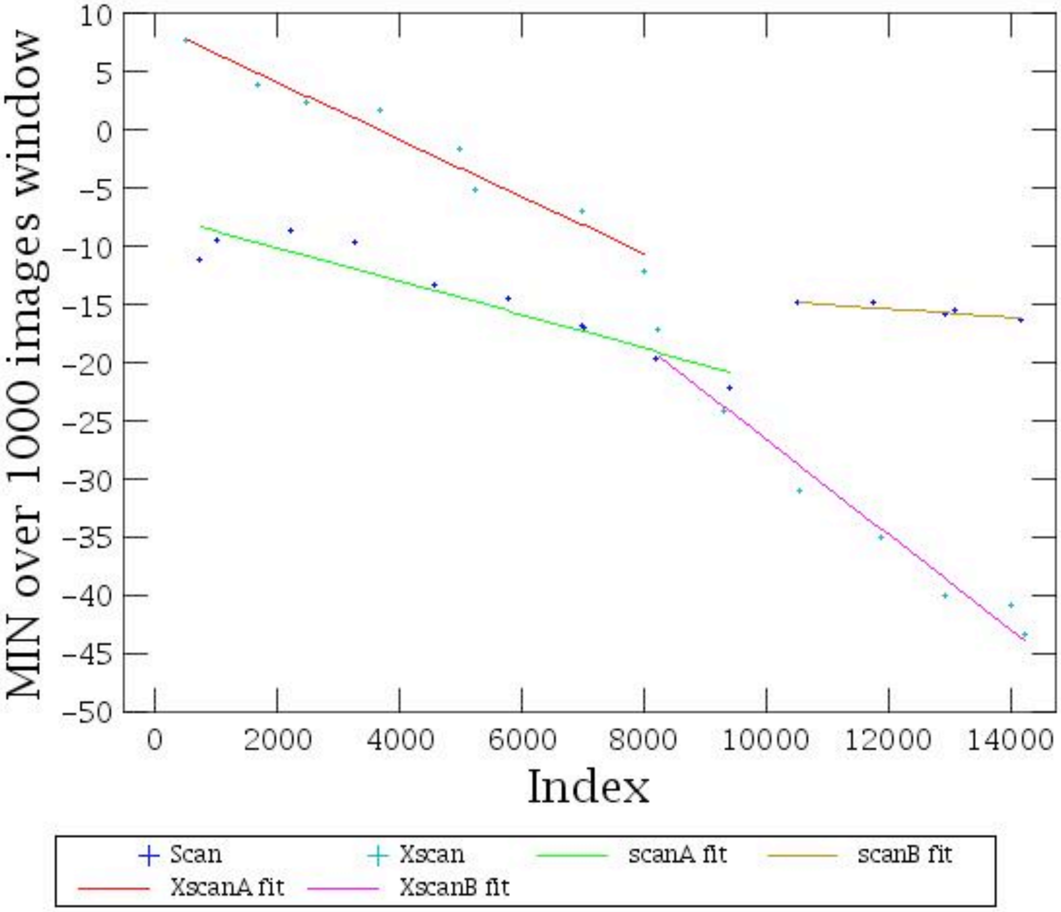


Figure 14 The minimum median (as described in the text) plotted versus its readout index. There also appears to be a change in the drift magnitude, likely caused by a cosmic ray or charged particle impact on the readout electronics. Hence, two best-fit lines are shown for each of scan and cross-scan observations.

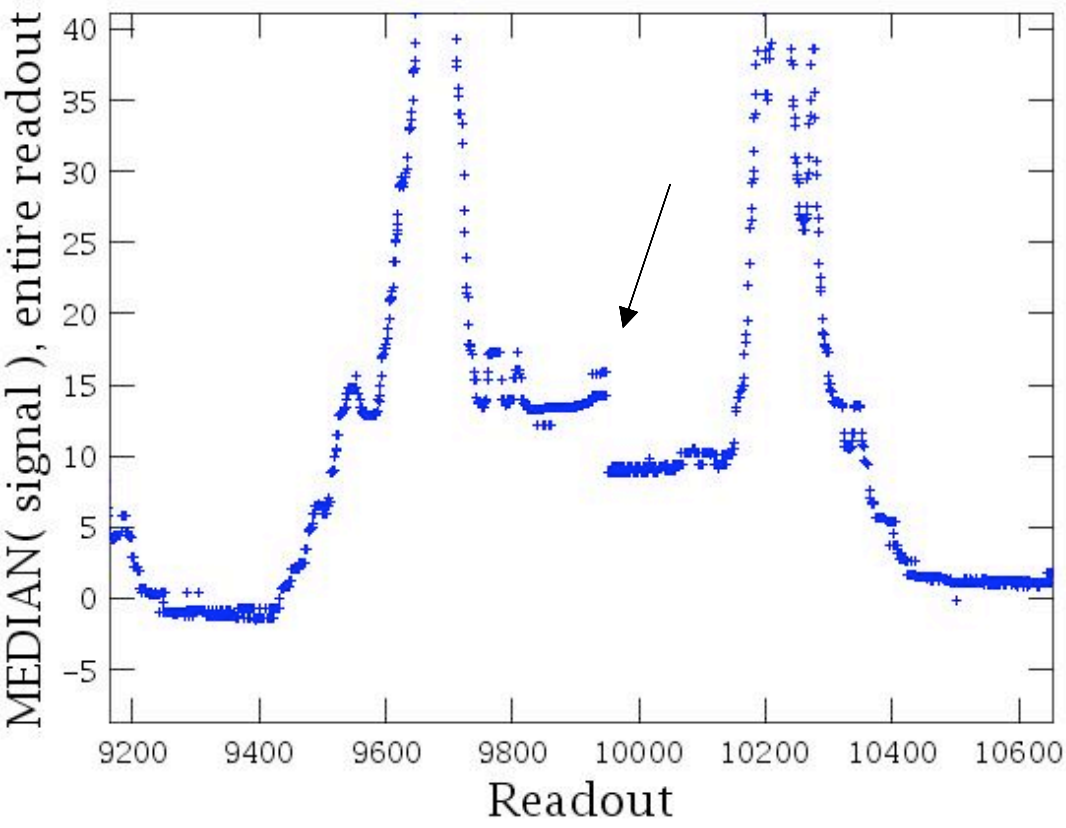


Figure 15 An expanded region of time-ordered data, near where the drift shows an abrupt change in magnitude in Figure 14. There is a clear break in the signal marked by the arrow.

8.2 Improper module-to-module drift correction.

If the inter-module drift is not corrected, the results will look similar to what is shown in Figure 16.

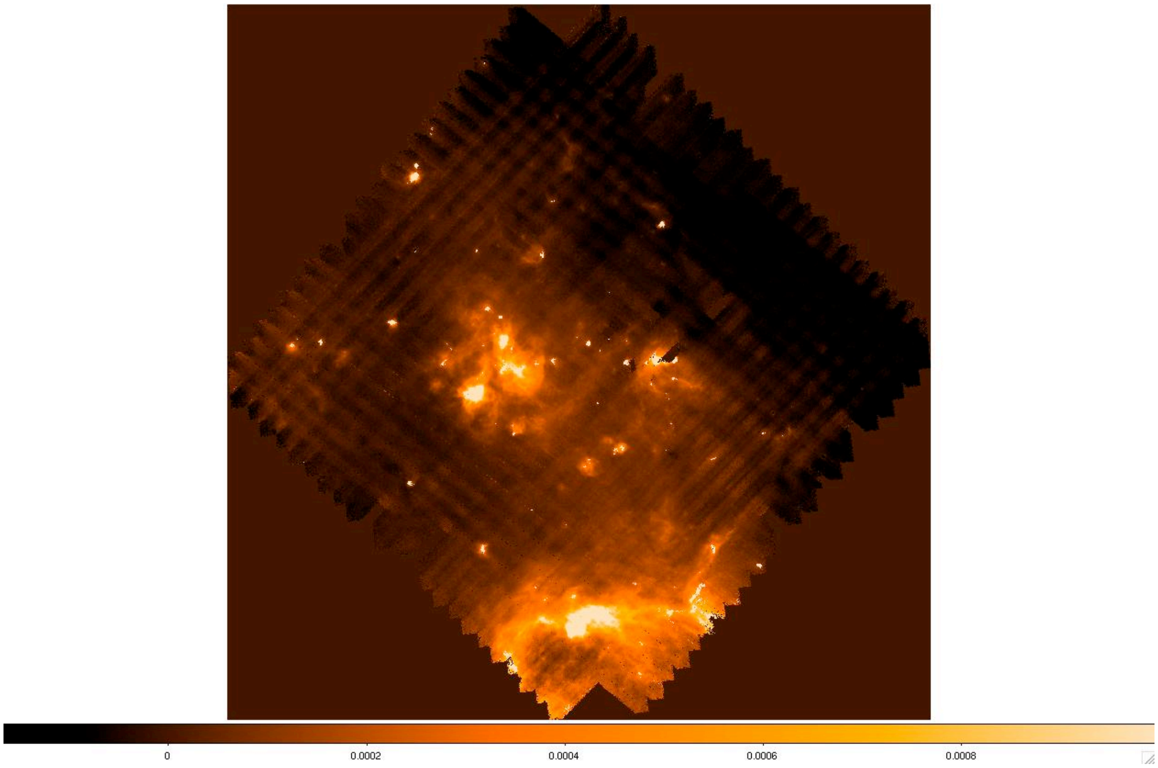


Figure 16 The final mosaic with a clearly visible "checkered" noise patter super-imposed on the sky. This artifact is due to improper correction for the module-to-module drift.

9 Bibliography

RD1	PICC-NHSC-TR-020	PACS InvNtt Calibration Files and Noise Analysis (D. Frayer, author)
RD2	PICC-NHSC-TR-028	PACS Photometric Check : MADmap vs Photproject (N. Billot, author)
RD3		Cantalupo, C. M., Borrill, J. D., Jaffa, A. H., Kisner, T. S., & Stompor, R. 2010, ApJS, 187, 212
RD4	MADmap UM	http://www.herschel.be/twiki/pub/Pacs/PacsMADmap/UM_MADmapDOC.txt