

NIRCam 1/f noise removal methods

Various methods have been developed to mitigate 1/f detector noise that appears as stripes in JWST data. None has yet been implemented in the JWST pipeline since detailed testing of these methods is still in progress, with [several of these methods being available publicly](#). Here we demonstrate these methods on NIRCam data of various scenes, both sparse and crowded. We find one method `image1overf` appears to work well in all cases. Further testing including quantitative analysis will be required before implementation in the JWST pipeline. Improved methods are also possible and under development.

- [1. Introduction](#)
- [2. 1/f software packages](#)
- [3. Data](#)
- [4. Results](#)
 - [4.1. Chris Willott's `image1overf`](#)
 - [4.2. Micaela Bagley's `remstripping`](#)
 - [4.3. Everett Schlawin's `ROEBA`](#)
 - [4.4. Dan Coe's `smooth1overf`](#)
 - [4.5. Massimo Robberto's `destriper`](#)
- [5. Detailed description of each method](#)
 - [5.1. Chris Willott's `image1overf`](#)
 - [5.2. Micaela Bagley's `remstripping`](#)
 - [5.3. Everett Schlawin's `ROEBA`](#)
 - [5.4. Massimo Robberto's `destriper`](#)
 - [5.5. Bernie Rauscher's `NSClean`](#)
 - [5.6. Dan Coe's `smooth1overf`](#)
- [6. Future work](#)

1. Introduction

JWST SIDECAR ASICs (detector readout electronics) generate significant 1/f noise during detector operations and signal digitization. This 1/f noise appears as horizontal or vertical banding that spans the entire width of the image, and vary from row to row or column to column depending on the employed read-out direction of all NIR JWST instruments. If not handled properly, it can cause one to overlook faint targets, or over-estimate the statistical significance of a detection on imaging data and introduce systematic errors in time series observations. More discussion of 1/f noise is provided on the [NIRCam TSO Caveats](#), [NIRCam Detector Performance](#), and [TSO Noise Sources pages](#).

The reference pixel correction step of the Stage 1 pipeline can partly mitigate 1/f noise. But, with the limited number of reference pixels available in NIRCam (and other JWST instruments) observations (four pixels on each edge of the full array images), the 1/f noise cannot be completely removed. 1/f correction is expected to be even less effective in subarray observations, which typically contain no or few reference pixels.

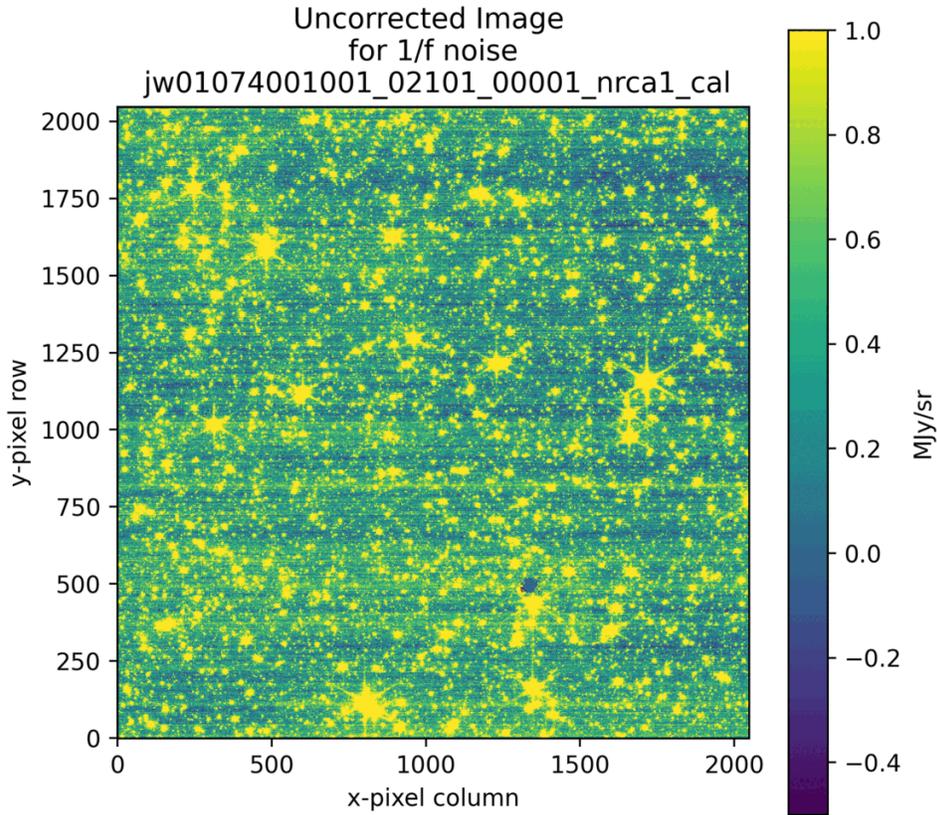


Figure 1. Demonstration of 1/f noise by blinking an uncorrected and corrected NIRC*am* image. The 1/f noise becomes evident as horizontal striping across the full detector.

2. 1/f software packages

Currently (as of September 2023) work is in progress to identify a suitably robust 1/f removal technique that can potentially be incorporated into the JWST pipeline in future, either for offline use (e.g., with manual tuning of parameters) or possibly for eventual automated use in operations. To date, various external software packages have been developed to identify and remove 1/f noise from JWST NIRC*am* imaging data. A non-comprehensive list of such packages that can be run on `*rate.fits` or `*cal.fits` files is listed in Table 1.

All methods are described in more detail below in Section 5. Note NSClean will be implemented in the JWST pipeline for NIRS*pec* data.

Table 1: 1/f-noise mitigation packages developed in Python for JWST imaging data. "Columns" and "Rows" refer to the fast and slow readout directions, respectively, though these directions may be swapped for different observing modes or instruments.

Package name	Developer	Rows (slow)	Columns (fast)	Full Rows	Individual Amplifiers	Retains Background	Source Masking	Notes
image1overf	Chris Willott	X		X	X	X	optional	Computes background image and employs it to remove row median levels
remstriping	Micaela Bagley	X	X	X	X		optional	Removes column and row medians from NIRC <i>am</i> rate or calibrated images
NSClean	Bernie Rauscher	X	X	X			user provided	Originally developed for NIRS <i>pec</i> MSA observations including many blank unilluminated pixels to precisely measure and subtract background Fourier modes
ROEBA	Everett Schlawin	X	X	X	X		user provided	Developed for NIRC <i>am</i> TSO weak lens NIRC <i>am</i> observations. Requires user-provided background image to remove column odd/even and row median levels.
smooth1overf	Dan Coe	X	X	X		X		Simple smoothing of sigma-clipped medians for rows and columns
destriper	Massimo Roberto	X			X	X		Remove median level per amplifier by employing the full rows

3. Data

The JWST NIRCam imaging mode is employed for a wide range of astronomical scenes, including sparse extragalactic fields, dense star clusters, and extended sources, such as diffuse and planetary nebulae. Below we consider a representative set of public data for various types of scenes, as summarized in Table 2 and Figure 2. All images have been calibrated using the STScI pipeline and obtained from the MAST archive. The pipeline processing includes the 1/f correction step using reference pixels; these are insufficient, leaving significant striping visible in the images.

Table 2. Data employed in our comparative analysis

Program identifier	Image identifier	Field type	Scene type
1345	jw01345001001_10201_00001_nrca3_cal.fits	Sparse	Extragalactic "blank" field
1074	jw01074001001_02101_00001_nrca1_cal.fits	Crowded	Star field
1334	jw01334001001_02101_00001_nrca1_cal.fits	Crowded	Dense star field
2107	jw02107025001_02101_00001_nrcb2_cal.fits	Extended 1	Galaxy close-up
2731	jw02731001001_02101_00004_nrca1_cal.fits	Extended 2	Nebula

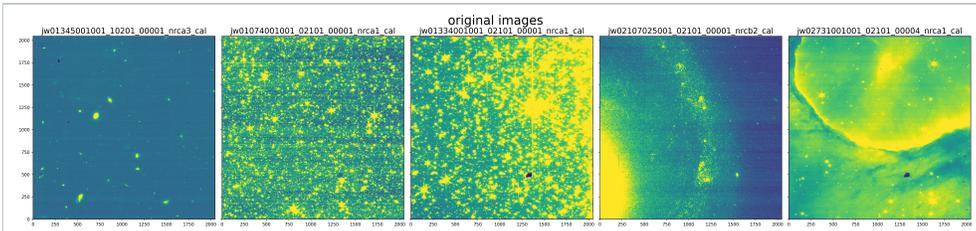


Figure 2. Images used in the comparative analysis

4. Results

All packages were run on the same data set to generate 1/f corrected images. Sources were detected and masked for codes allowing that option. All results are shown below on linear scales, clipped to the same flux levels.

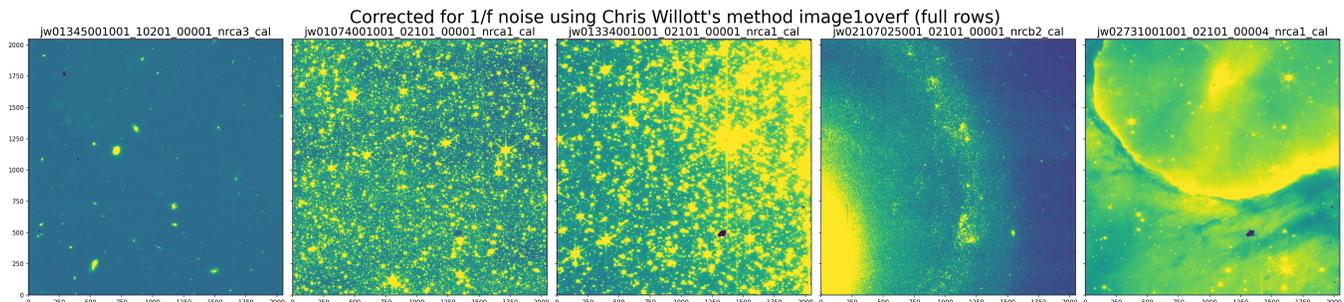
We emphasize most methods were not designed for these general cases. Some assume sparse scenes with "blank" data regions available to measure the 1/f noise. In these cases, all of the methods perform reasonably well. When the image is more crowded or full, most methods do not perform well, oversubtracting the background and/or introducing discontinuities. This can impact the accuracy of absolute flux measurements.

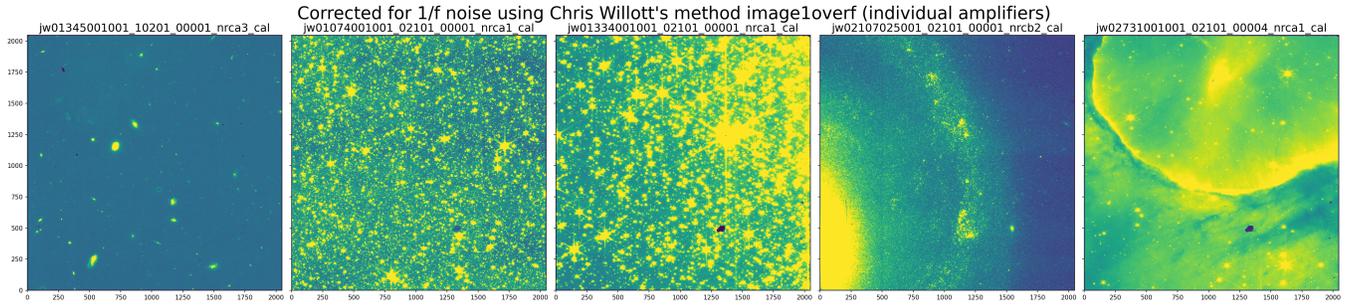
The parameters of some of the packages can be optimized depending on the scene to be corrected. An exhaustive exploration of all options for improving the parameters is beyond the scope of this article.

Package versions may change owing to method improvements and bug fixes. To avoid version mismatch issues, each of the Jupyter Notebooks linked to this article also provide working versions of the 1/f noise mitigation packages employed for this analysis. It is left at the discretion to each user to adjust the latest up-to-date package version and identify potential differences from the results presented here.

4.1. Chris Willott's image1overf

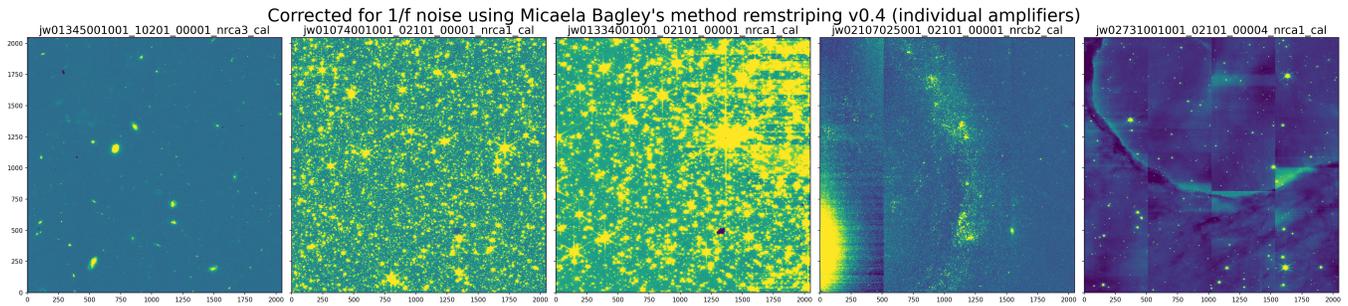
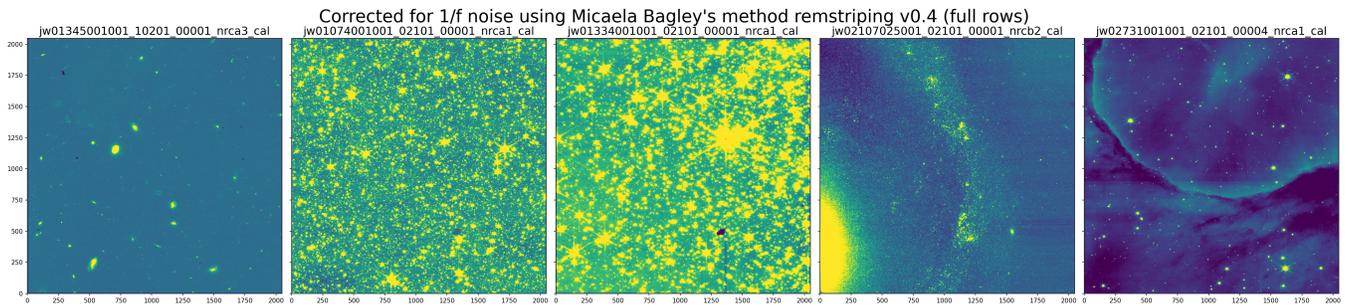
We find this method performs well for all scenes tested, including preserving the overall flux levels without oversubtracting "background". The individual amplifier option works well on the sparse fields but introduces slight artifacts in the full fields. The code can also be run without source masking; the results are nearly identical to those with it.





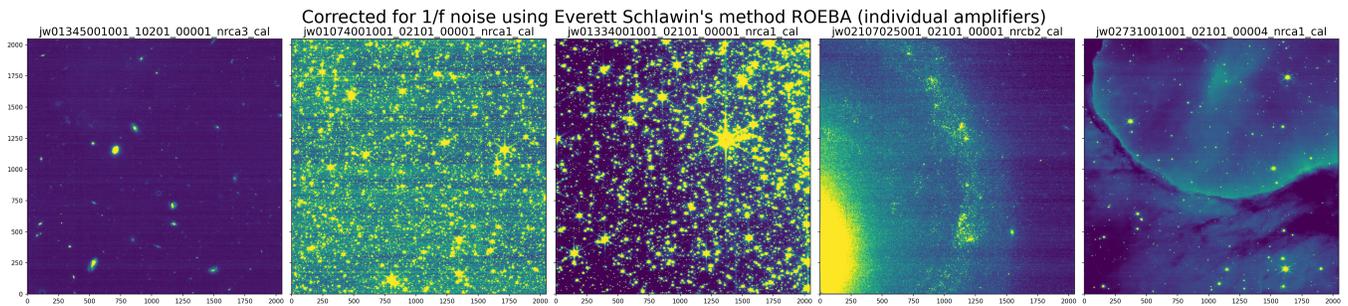
4.2. Micaela Bagley's remstriping

Using [v0.4 available here](#), which fixes a few bugs. Originally developed for the CEERS program, this method performs well on the image from that dataset as well as the sparse star field. Since it measures and subtracts the 2D background, fuller fields are left oversubtracted. Artifacts are introduced when correcting amplifiers individually.



4.3. Everett Schlawin's ROEBA

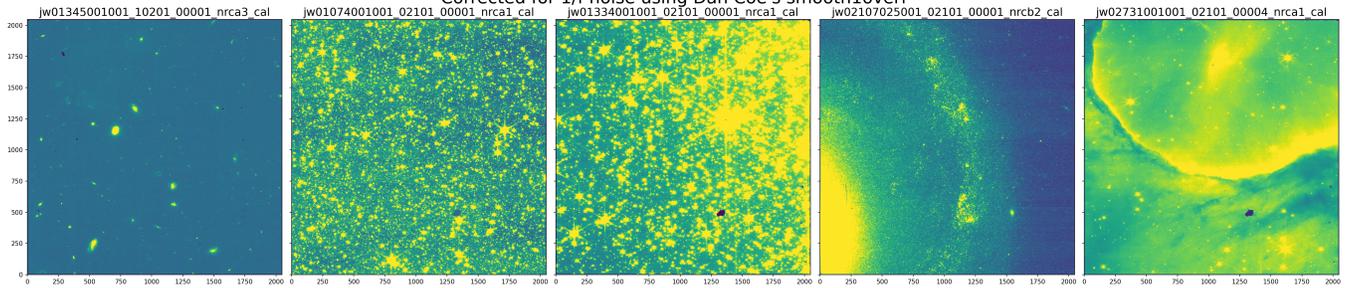
Using the default parameters, this method oversubtracts background while leaving some faint striping.



4.4. Dan Coe's smooth1overf

Designed to simply smooth the row and column medians, this method performs well in the general case, while leaving some residual striping, especially on larger scales (more than a few rows).

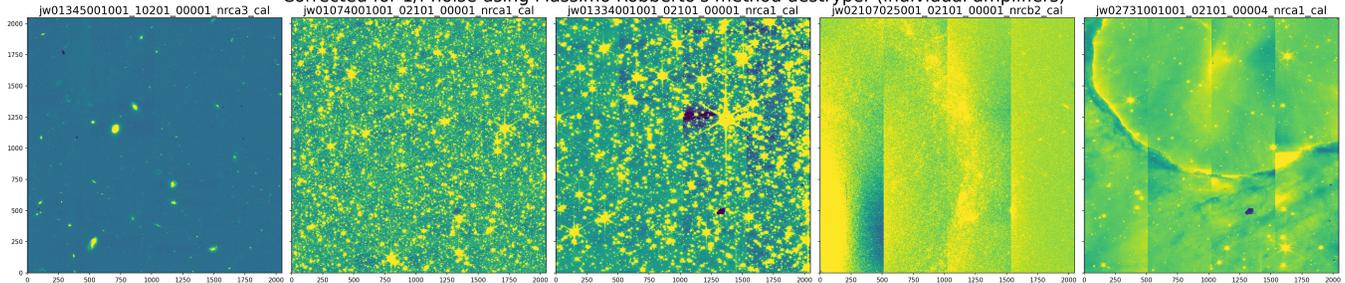
Corrected for 1/f noise using Dan Coe's smooth1overf



4.5. Massimo Robberto's destriper

Another simple method, this median smoothing works well on sparse fields. By processing amplifiers individually, artifacts are introduced in more full scenes.

Corrected for 1/f noise using Massimo Robberto's method destriper (individual amplifiers)



5. Detailed description of each method

Each of the 1/f noise mitigation packages reviewed in this article remove median image background. The packages also share the assumption for a negligible variation of the 1/f noise between two neighbor amplifiers, and a sufficient background region without scattered light or faint background point sources. The following test results include also imaging fields of extended sources as well as crowded star fields with the goal to demonstrate the performance of the individual packages including beyond their assumptions.

5.1. Chris Willott's image1overf

Example Jupyter Notebook for running image1overf: [from Chris Willott](#), or [from STScI](#)

General description and assumptions: Performs a removal of 1/f readout noise on NIRISS or NIRCcam level 2 calibrated image by masking objects and subtracting the median of each column. The correction method assumes that there isn't a significant variation in the background across the field. The code has been designed to handle 1/f stripes either assuming a constant noise along the full detector rows, or amplifier-to-amplifier 1/f noise differences. Detector orientation is determined automatically by searching for the value of the header keyword 'SLOWAXIS'. Only two subarrays are supported: FULL (NIRCcam and NIRISS) and SUB256 (NIRISS) subarrays. It includes the effect of a variable background (by temporarily removing it before determining the 1/f correction and adding it back afterwards) and masking pixels containing sources. The author emphasizes that the code should be used with care and the users need to inspect the results for any unintended consequences.

Difference compared to all other packages: By temporarily subtracting a 2D background to measure and subtract stripes, the pedestal flux level of the image is preserved, which is especially helpful for crowded scenes.

Method steps:

1. Subarray type is determined and non-reference pixels are extracted
2. Mask bad pixels, determined from the DQ array
3. Mask all pixels outside ± 3 -sigma of global median (adjustable with `sigma_bgmask` parameter)
 - a. assumes 1/f noise variations are within this 3-sigma; also assumes background doesn't vary significantly across the field
4. Temporarily subtract 2D background grid measured using photutils
5. Mask additional pixels one of two ways:
 - a. Detect sources using photutils
 - b. Mask all pixels outside ± 2 -sigma of global median of background-subtracted image (adjustable with `sigma_1fmask` parameter)
6. Measure row stripes: medians of remaining unmasked pixels
 - a. Perform within each amplifier individually (ABCD) or across full rows (controlled by `splitamps` parameter)
 - b. Only dominant stripes are corrected, usually rows (sometimes columns) depending on which detector axis is read out slowest (determined from FITS header keyword `SLOWAXIS`)
7. Subtract row stripes from data (without background subtraction)

Source code and documentation:

- <https://github.com/chriswillott/jwst>
- <https://github.com/chriswillott/jwst/blob/master/image1overf.py>

5.2. Micaela Bagley's remstriping

Example Jupyter Notebook for running remstriping: [from STScI](#)

General description and assumptions: Designed for sparse extragalactic "blank" fields observed by programs like [CEERS](#), remstriping measures both row and column striping

For DR0.5, we visually inspected each $1/f$ noise-subtracted image and adjusted the masking threshold for the images with over- or under-subtracted pattern noise. For example, images with large, bright sources usually benefit from a threshold closer to 0.1 than 0.8. The threshold can either be changed using the global variable MASKTHRESH, or by setting a value manually for each image"

Originally designed to be run on rate.fits files, remstriping includes a step to perform flat fielding before $1/f$ measurement & correction. The remstriping algorithm can also be run on cal.fits files by turning off the flat fielding step.

Difference compared to all other packages:

Method steps:

1. Apply flat field (optional, assuming running on rate files)
2. Mask sources detected using photutils 1.5 (available option to load source mask)
3. Subtract median (single value, either simple median or mean of Gaussian fit)
4. Measure row and column striping: sigma-clipped median of all unmasked pixels
 - a. Measure within each amplifier individually (ABCD) and also across full rows
 - b. Use individual amplifier measurements, unless most of the pixels are masked, then revert to the full row/column measurements
 - c. Controlled by threshold parameter `thresh`: default requires 80% of pixels to be good (unmasked and free of detected sources)
5. Subtract stripes from data (including background subtraction, without flat field correction)

Source code and documentation:

- <https://github.com/ceers/ceers-nircam/blob/main/stage1.md>
- <https://github.com/ceers/ceers-nircam/blob/main/remstriping.py>
- <https://github.com/ceers/ceers-nircam/blob/main/README.md#requiredpackages>

5.3. Everett Schlawin's ROEBA

Example Jupyter Notebook for running ROEBA: [from STScI](#)

General description and assumptions: [ROEBA](#) stands for Row-by-row, Odd/Even By Amplifier correction and is part of the standalone package for reduction of transiting exoplanet data called [tshirt](#). ROEBA has been developed for removal of $1/f$ noise in the SW channel of NIRCcam TSOs (which can differ significantly compared to imaging fields, [Schlawin et al. 2021, 2023](#)). Such scenes typically consist of a bright source, whose PSF is defocused using one of the NIRCcam SW weak lenses, and usually dominates the scene (in non-crowded fields) with a free of bright sources field providing plenty of background. In order to run ROEBA's correction, the user needs to provide a background region to employ as a proxy for reference pixels that have minimal contamination from bright sources. The package relies on the assumption for a clean background region and could potentially over-subtract light or background sources where this condition is not met.

Difference compared to all other packages: Requires user-provided background mask. Corrects for odd/even effect and uses row medians from each amplifier, or full row to correct for $1/f$ noise.

Method steps:

1. By employing a user-provided background image, ROEBA applies correction in the slow-read-out direction, which subtracts the median odd count level from all odd-numbered columns and the median count level from all even-numbered columns; this first step removes the pre-amplifier offsets;
2. Calculates row-by-row median levels from the background region and subtracts this from all pixels in the given row; this reduces the $1/f$ noise
 - a. controlled by `amplifiers` parameter

Source code and documentation:

- <https://github.com/eas342/tshirt>
- https://tshirt.readthedocs.io/en/latest/specific_modules/ROEBA.html#Row-by-row,-Odd/even-by-amplifier-correction
- [Schlawin, E., et al. 2023, PASP, 135, 8001](#)

5.4. Massimo Robberto's destriper

Example Jupyter Notebook for running destriper: [Massimo's destriper V2.ipynb](#), or [from STScI](#)

General description and assumptions: The destriper package takes as input calibrated or rate NIRCcam images, and corrects for $1/f$ noise per amplifier by removing the median level of each row. There is no assumption for the user to provide background regions for determining the median for each row.

Difference compared to all other packages: Relies on median level of amplifier image rows without source masking.

Method steps (per individual NIRCam amplifier):

1. Extract image area without the reference pixels from the science extension
2. Compute median level for each row
3. Subtract the median for each row and offset to the median of the input data
4. Subtract median for each amplifier and offset to the median level of the first amplifier

Source code and documentation:

- Jupyter Notebook provided by Massimo Robberto: [Massimos destriper V2.ipynb](#)

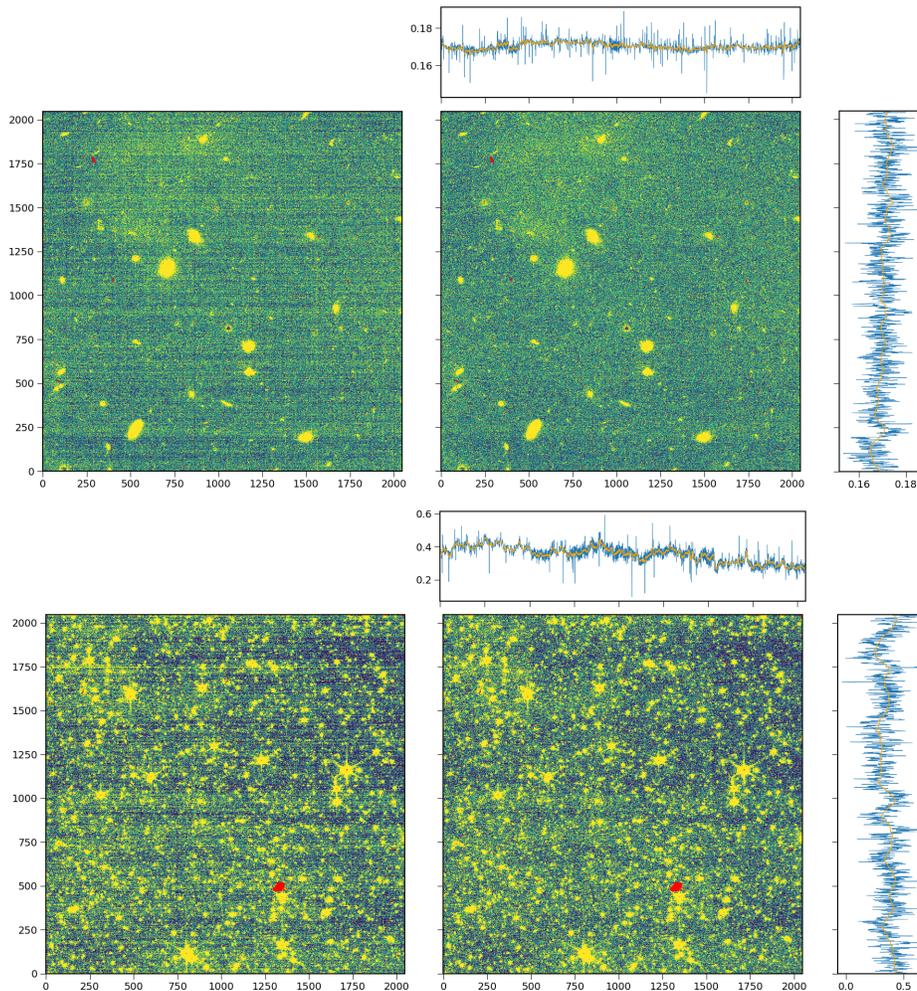
5.5. Bernie Rauscher's NSClean

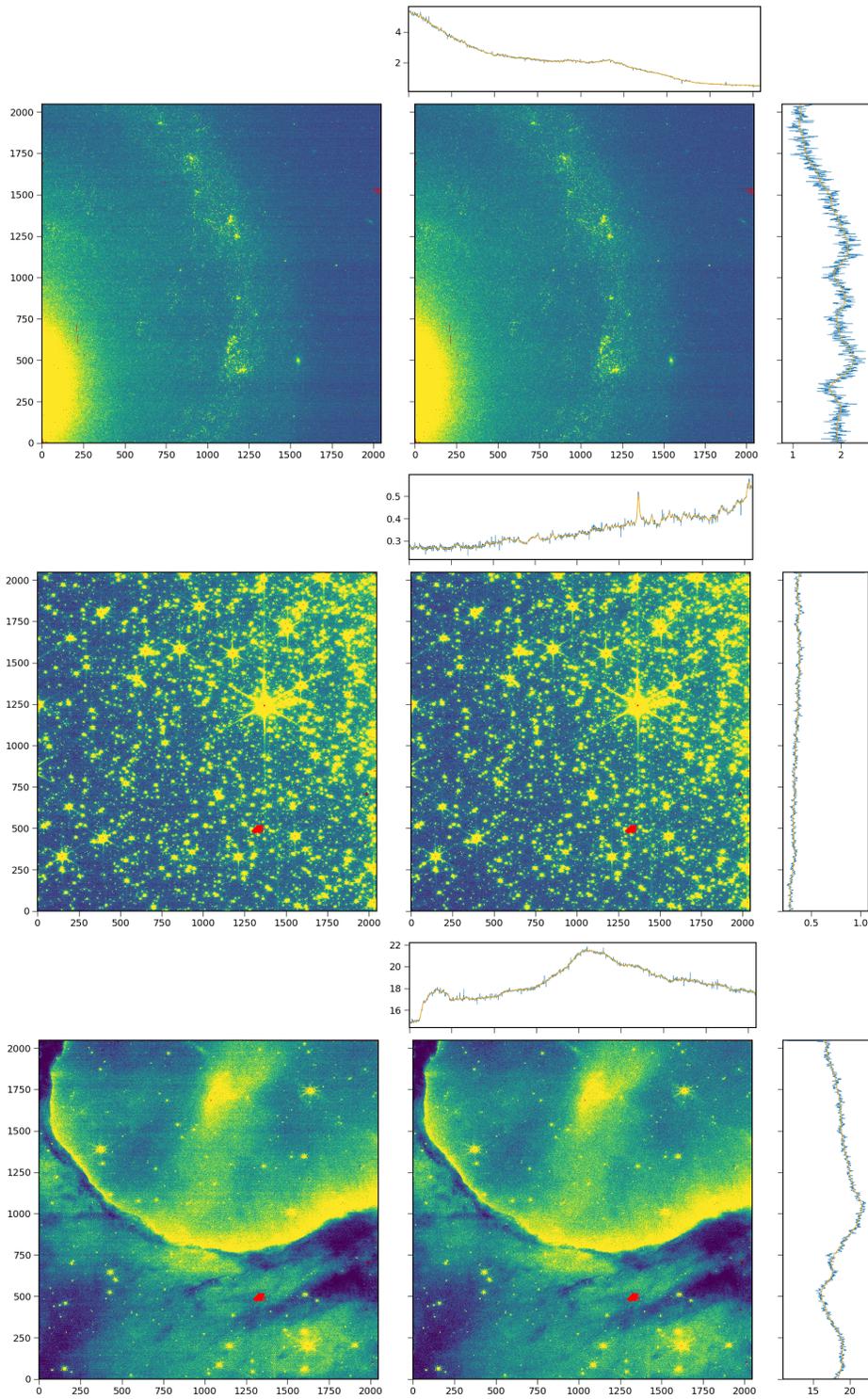
Originally developed for NIRSpec MSA observations including many blank unilluminated pixels to precisely measure and subtract background Fourier modes, using a robust theoretical basis. It can also be applied to NIRCam images of individual objects (coronagraphy or TSO) surrounded by many "blank" pixels. It was not designed for the scenes tested above, containing many objects. In those cases, additional Fourier modes are detected, measured, and subtracted, leaving artifacts, even in "blank" fields.

Work is underway to implement NSClean in the JWST pipeline for NIRSpec data.

5.6. Dan Coe's smooth1overf

This simple method calculates sigma-clipped medians of each row / column, then smooths them with simple rolling averages. Below we show original (left) and corrected (right) images along with the sigma-clipped medians of columns (top) and rows (right) in blue, and the smoothed medians in orange used to correct the images.





6. Future work

Additional work is required to test methods against more datasets and perform quantitative analysis before considering implementation into the JWST pipeline.

Additional development of methods and code is also underway by various groups. For example, $1/f$ noise is present in every detector read and should be subtracted from individual reads in the `uncal.fits` files before (or as part of) ramp fitting to generate the `rate.fits` files.