

2
3
4
5
6
DRAFT (Rev. 1.5)
NSClean: An Algorithm for
Removing Correlated Read Noise from JWST NIRSpec Images

BERNARD J. RAUSCHER ¹

¹*NASA Goddard Space Flight Center*
Observational Cosmology Laboratory
8800 Greenbelt Road
Greenbelt MD 20771, USA

7 Cite as Rauscher, B.J. (2023), PASP, in preparation

8 **ABSTRACT**

9 NSClean is an algorithm and associated python package for removing faint vertical
10 banding and “picture frame noise” from JWST Near Infrared Spectrograph (NIRSpec)
11 images. NSClean uses known dark areas to fit a background model to each exposure
12 in Fourier space. When the model is subtracted, it removes nearly all correlated noise.
13 Compared to simpler strategies like subtracting the rolling median, NSClean is more
14 thorough and uniform. NSClean is computationally undemanding, requiring only a few
15 seconds to clean an image on a typical laptop. The NSClean package is freely available
16 for download from the NASA JWST website ([NASA JWST website 2023](#)).

17 **1. INTRODUCTION**

18 JWST is today’s premier observatory for mid
19 and near-infrared (NIR) space astronomy. To
20 enable science objectives cutting across astro-
21 physics, JWST carries a suite of four science
22 instruments: a Near Infrared Camera (NIR-
23 Cam; Rieke et al. 2022), a Near Infrared Imager
24 and Slitless Spectrograph (NIRISS; Doyon et al.
25 2022), a Mid-infrared Instrument (MIRI; Rieke
26 et al. 2022), and a Near Infrared Spectrograph
27 (NIRSpec; Jakobsen et al. 2022). This article
28 concerns NIRSpec, an algorithm and associated
29 software for further reducing its already low
30 noise: “NSClean”. NSClean should be benefi-

31 cial to most NIRSpec Integral Field Unit (IFU)
32 and many Multi-Object Spectrograph (MOS)
33 observers.

34 From early on, it was understood that NIR-
35 Spec required ultra-low noise detectors. It is de-
36 tector noise limited for all but prism-mode ob-
37 servations. This is in contrast to other JWST
38 instruments that are generally limited by the
39 astronomical background. Consequently, NIR-
40 Spec has lower noise requirements than other
41 JWST instruments. “Total noise” is a con-
42 cept that was introduced for JWST. To measure
43 it; one defines a standard scientific exposure,
44 takes many such exposures (typically >40), and
45 then computes the standard deviation per pixel.
46 Across JWST’s NIR instruments, the exposure
47 time was taken to be 1000 seconds. For NIR-
48 Cam and NIRISS, median total noise was re-

49 quired to be $<10 e^-$ per exposure. For NIR-
50 Spec, the requirement was $<6 e^-$.¹

51 NIRSpec’s $<6 e^-$ noise requirement is the rea-
52 son why we developed Improved Reference Sam-
53 pling and Subtraction (IRS²; pronounced IRS-
54 square; Rauscher et al. 2017). In IRS² mode,
55 NIRSpec uses a special clocking pattern and ref-
56 erence correction pipeline step to reduce corre-
57 lated noise as far as possible using the NIRSpec
58 detector’s built-in references. Using IRS², NIR-
59 Spec’s total noise is slightly $<6 e^-$ on average,
60 and to within the uncertainties compliant with
61 requirements. IRS² is the recommended read-
62 out mode for most observations except for ex-
63 tremely bright targets (JWST User Documen-
64 tation website 2016).

65 However, even with NIRSpec’s detectors meet-
66 ing requirements, many NIRSpec observers re-
67 port seeing faint, correlated read noise in count
68 rate images that complicates calibration. For-
69 tunately, for NIRSpec, much of this can be re-
70 moved by using dark areas of images as refer-
71 ences.

72 Figure 1 shows an example of the correlated
73 noise from an early NIRSpec Integral Field Unit
74 (IFU) observation. We have smoothed the im-
75 ages and stretched the greyscales to emphasize
76 correlated noise that would otherwise be more
77 difficult to see against the background of NIR-
78 Spec’s ~ 6 electrons total noise. One sees a
79 “picture frame” effect, whereby areas near the
80 edges of both detectors on all four sides seem
81 less noisy. In the interiors, one sees faint verti-
82 cal striping. While the amplitude is small, this
83 correlated noise can undermine accurate pho-
84 tometry when no local sky is available. This is
85 often the case for IFU observations and we are
86 aware of cases where this is true also in MOS
87 mode.

¹ MIRI uses a different detector technology for which the
comparison is not relevant.

88 NSClean uses blanked off areas of NIRSpec
89 scenes to model the background, including cor-
90 related noise.

91 Because it uses more information, NSClean’s
92 correlated noise correction is more complete and
93 more uniform than is possible without careful
94 masking.

95 2. PHYSICAL CAUSE OF THE 96 CORRELATED NOISE

97 Our focus in this paper is on the specific corre-
98 lated read noise that NSClean is designed to fix.
99 Readers who want to learn more about NIR-
100 Spec’s read noise in general may want to see
101 some of our earlier papers. Rauscher (2015)
102 describes the origins of NIRSpec’s white and
103 $1/f$ noise, and provides a python package for
104 simulating it. Rauscher et al. (2017) describes
105 NIRSpec’s IRS² readout mode. Without IRS²,
106 the residual correlated noise that remains today
107 would be much worse.

108 The correlated noise that remains after IRS² is
109 a logical consequence of how IRS² works. NIR-
110 Spec uses two Teledyne H2RG NIR detector ar-
111 rays (Loose et al. 2003). Each H2RG provides
112 two types of reference information that can be
113 used to remove correlated read noise. These
114 are the “reference pixels” that form a 4-pixel
115 wide frame on all sides of NIRSpec images and
116 one “reference output” per H2RG. The refer-
117 ence output is not visible in the usual pipeline
118 data products, but it is used most of the time.
119 As described in Rauscher et al. (2017), IRS²
120 is built on principal component analysis (PCA)
121 showing that NIRSpec’s read noise is covariance
122 stationary to a high degree of approximation.
123 Informally, this means that the read noise is in-
124 dependent of when one looks.

125 It turns out that in JWST’s NIR detector sys-
126 tems, thermal instability causes noise that is
127 not covariance stationary. There is a picture
128 frame pattern that changes in time at the $\sim 1 e^-$
129 level. Rauscher et al. (2013) describe how small
130 temperature fluctuations can drive the picture

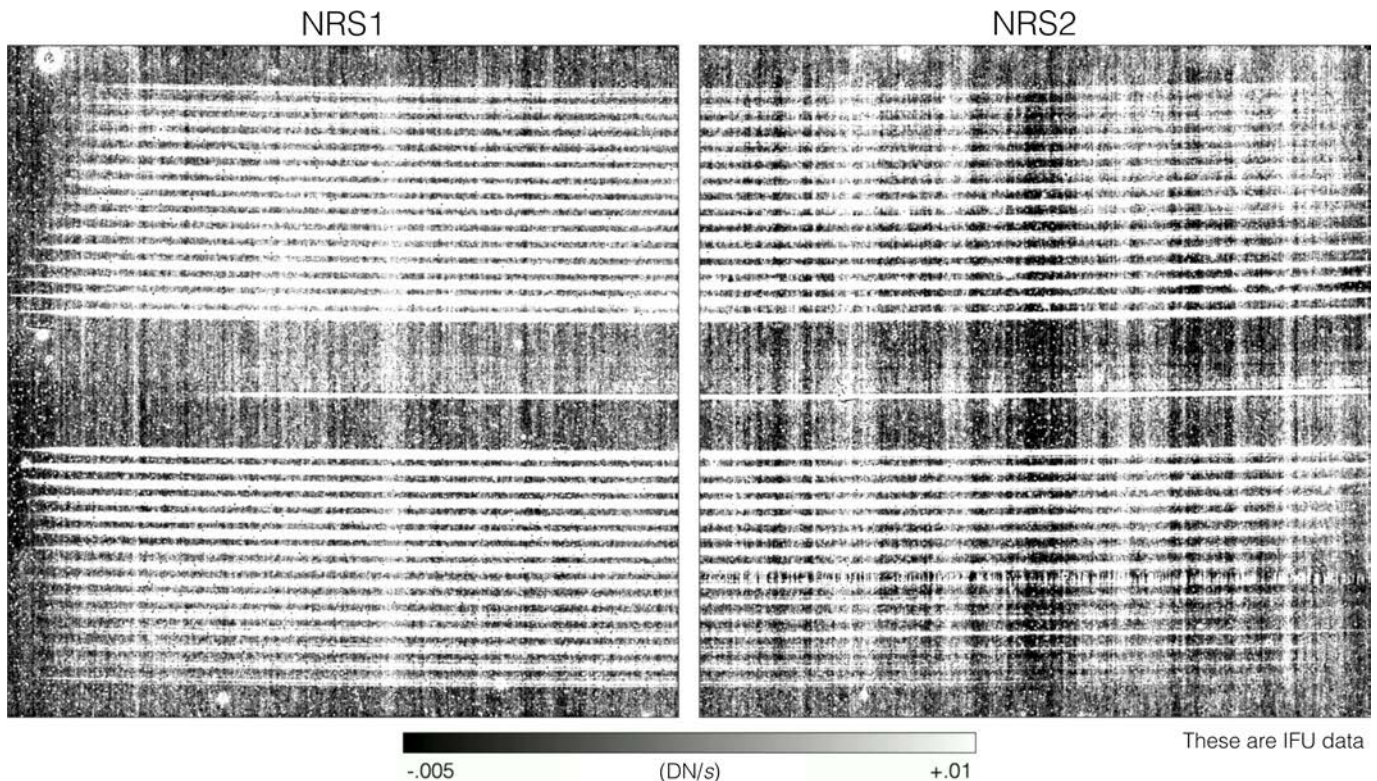


Figure 1. The JWST pipeline makes NIRSpect count rate images like those shown here. This observation used NIRSpect’s IFU mode which produces 30 horizontal spectral traces per detector. To highlight correlated noise, we have smoothed the images and set the greyscale roughly equal to NIRSpect’s 6 electrons total noise requirement. One sees vertical banding in the central regions of both detectors. Toward the edges of both detectors, there seems to be less correlated noise. This is the “picture frame”. While both types of residual noise are less than NIRSpect’s total noise requirement, they nevertheless complicate calibration. For example, they can produce negative fluxes and features that mimic emission lines or continuum. NSClean fits a background model to dark areas of each exposure and subtracts it to remove picture frame noise and vertical banding.

131 frame. This is why the vertical banding that
 132 is visible in Figure 3a seems to fade away near
 133 the edges. The relatively quiet edges are in the
 134 picture frame while the vertical bands are not.
 135 IRS² relies on the reference pixels to see noise
 136 in order to remove it. Since the reference pix-
 137 els are in the picture frame and do not see the
 138 vertical banding, IRS² is powerless to remove it.

139 3. ALGORITHM

140 NSClean is built on the Fourier transform of
 141 the instrumental background. Our treatment
 142 starts in Section 3.1, by reviewing how python’s
 143 numpy package implements the classical Fast
 144 Fourier Transform (FFT; Cooley & Tukey 1965)

145 for fully sampled data. Since NIRSpect’s back-
 146 ground is not fully sampled (because of as-
 147 tronomical sources), Section 3.2 explains how
 148 NSClean computes a statistically optimal ap-
 149 proximation to the Fourier transform using all
 150 available background samples.

151 The next two subsections describe the linear
 152 algebra that underpins NSClean. Insofar as pos-
 153 sible, we have tried to use a consistent, standard
 154 notation. Throughout this paper, boldface low-
 155 ercase letters are vectors and uppercase boldface
 156 letters are matrices. When discussing matrix el-
 157 ements, we use superscripts for row indices and
 158 subscripts for column indices.

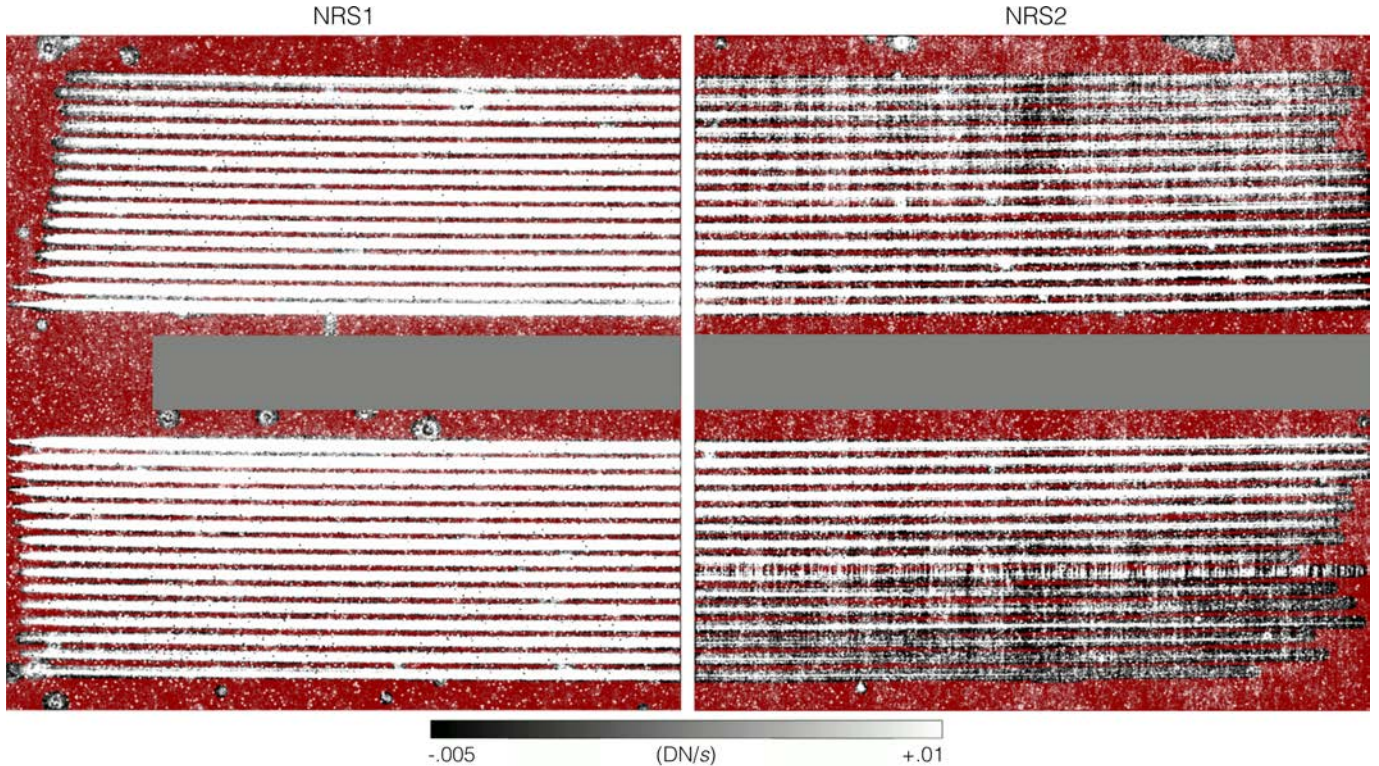


Figure 2. We used the background masks shown here for development. The underlying grayscale image is the median of a stack of illuminated IFU exposures. The 30 spectral traces per detector are clearly visible. We used the red-shaded pixels to make the background model. As described in the text, we used the GNU Image Manipulation Program (GIMP) to manually make the masks since we only needed one set to write the software. We understand that some JWST observers have already automated mask generation. They grey rectangles blank off areas of potentially illuminated (by scattered light) areas of the focal plane that we left unconstrained during background fitting.

3.1. Numpy's Classical FFT

For dark exposures, one can use numpy's FFT package to compute the Fourier transform of an image column. Like all FFTs, numpy uses a highly-efficient factorization of the Fourier matrix, \mathbf{F} , to solve the matrix equation,

$$\mathbf{F}\mathbf{f} = \mathbf{d}, \quad (1)$$

where \mathbf{f} is the Fourier transform of the data, \mathbf{d} . For n pixels per column, in numpy the elements of \mathbf{F} are,

$$F_k^m = \exp\left\{2\pi i \frac{mk}{n}\right\}, \quad (2)$$

where m is the row index and k is the column index. Because NIRSpec's data are real valued and $n = 2048$ is an even number; $m = 0, 1, \dots, n - 1$ and $k = 0, 1, \dots, n/2$.

3.2. NSClean's Fourier Transform

For NIRSpec's incompletely sampled background, NSClean uses weighted least squares to approximate Fourier transforms. The starting point is again equation 1,

$$\mathbf{F}\mathbf{f} \approx \mathbf{d}, \quad (3)$$

but now as an approximation and with the understanding that \mathbf{F} , \mathbf{f} , and \mathbf{d} are incomplete. \mathbf{F} is missing columns where light falls on the detector and rows for frequencies that we choose not to fit. \mathbf{f} contains only a few very low frequencies to minimize noise. \mathbf{d} is missing rows where the detector is illuminated.

To solve equation 3 using least squares, we minimize the generalized distance squared,

$$\delta^2 = (\mathbf{F}\mathbf{f} - \mathbf{d})^H \mathbf{W} (\mathbf{F}\mathbf{f} - \mathbf{d}), \quad (4)$$

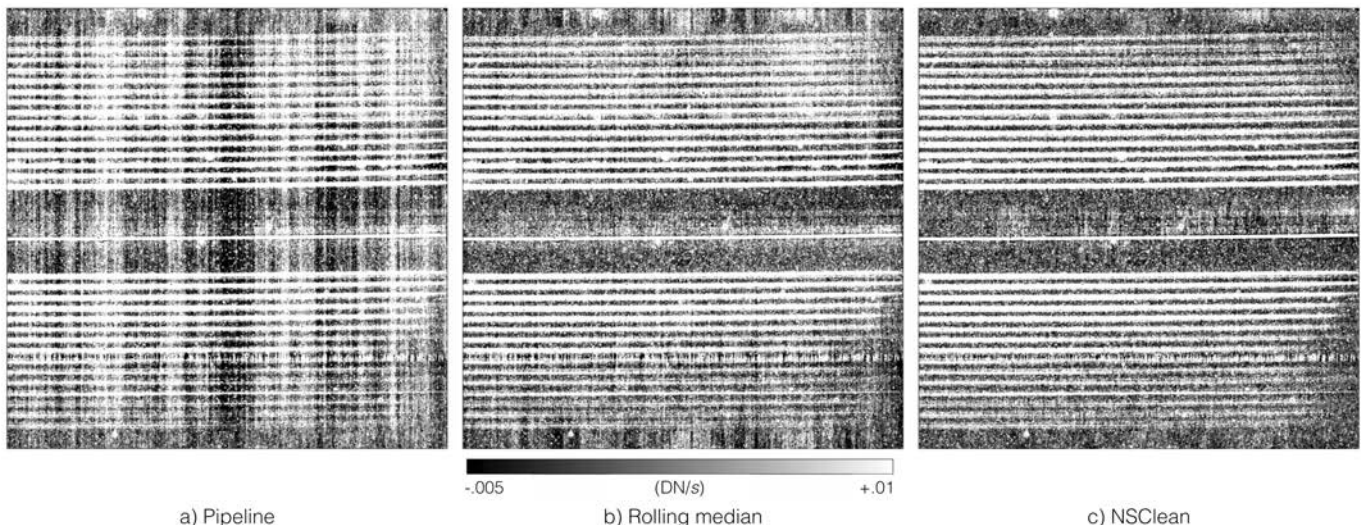


Figure 3. This figure shows the a) correlated noise that is visible in pipeline calibrated images. The actual pipeline products do not look this bad. We have adjusted the grayscale and blurred the images slightly to highlight correlated read noise. Panel b) shows the effect of subtracting the median of a few neighboring columns from each column. The NIRSspec Instrument Team previously provided a tool to NIRSspec observers that does this. Finally, panel c) shows the NSClean result. Panels b and c are noticeably cleaner than panel a. Comparing panels b and c, panel c shows more uniform and complete background subtraction.

190 using all available background samples. The
 191 symbol, “ H ”, denotes the conjugate transpose,
 192 which is also known as the Hermitian transpose.
 193 A weight matrix, \mathbf{W} , is required to compensate
 194 for non-uniform background sampling. The cur-
 195 rent version of NSClean weights inversely by the
 196 local sample density squared, ρ^{-2} :

$$197 \quad \mathbf{W} = \begin{bmatrix} \rho_{00}^{-2} & 0 & 0 & 0 \\ 0 & \rho_{11}^{-2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \rho_{n'-1}^{-2} \end{bmatrix}. \quad (5)$$

198 \mathbf{W} is diagonal and equal to it’s conjugate trans-
 199 pose. Section 3.3 describes \mathbf{W} in more detail.
 200 The quantity $n' \leq n$ is equal to the number of
 201 background samples. Under these conditions,
 202 the least squares solution to Equation 4 is,

$$203 \quad \mathbf{f} = (\mathbf{W}^{1/2}\mathbf{F})^+ \mathbf{W}^{1/2}\mathbf{d}. \quad (6)$$

204 The symbol, “ $+$ ”, denotes the Moore-Penrose
 205 inverse. Being a Fourier transform, the quantity
 206 \mathbf{f} is a complex valued vector.

207 Equation 6 is this section’s key result.
 208 NSClean uses this expression to approximate
 209 the Fourier transform of the incompletely sam-
 210 pled background.

211 Figure 4 shows an example of how equation 6
 212 works in practice. Panel a) shows a vertical cut
 213 through NRS2, which is the most affected of the
 214 two detectors. To show detail, Panel b) shows
 215 only the innermost 1024 rows. The blue points
 216 are background samples, the orange points are
 217 pixels that the background mask marked as po-
 218 tentially illuminated, and the blue line is the
 219 model built using equation 6. As a practical
 220 matter, we were able to fit about nine frequen-
 221 cies (≈ 16 free parameters) before we started
 222 to see increased noise due to over fitting. As
 223 expected, the blue line passes near the centers
 224 of groups of blue points. It is smooth, continu-
 225 ous, and very low noise compared to the pixels
 226 themselves.

3.3. The Weight Matrix, \mathbf{W}

227 The weight matrix compensates for uneven
 228 background sampling. Returning to Figure 2,

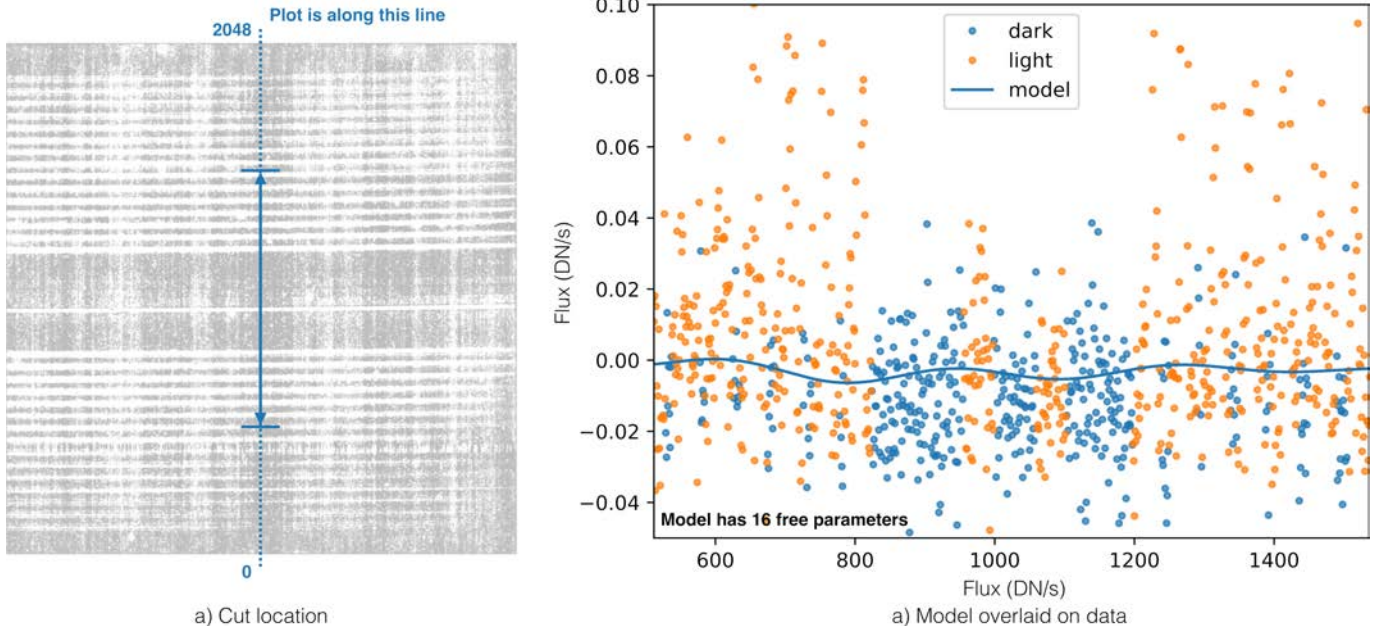


Figure 4.

230 there are often only a few rows of blanked off
 231 background pixels between the spectral traces.
 232 But; near the bottom, middle, and top of each
 233 detector, there are much larger areas of back-
 234 ground pixels. When nothing is done to com-
 235 pensate for the uneven background sampling,
 236 scientifically uninteresting areas of the scene
 237 carry far too much weight.

238 As described earlier, NSClean computes the
 239 Fourier transforms of columns individually us-
 240 ing weighted least squares fits. After a bit of
 241 trial and error, we found that weighting in-
 242 versely by the local background sample density
 243 in columns worked well. There is nothing funda-
 244 mental about this weighting scheme. We imag-
 245 ine that some observers will find better ones for
 246 their data.

247 One could compute the local sample den-
 248 sity by convolving the background mask with a
 249 tophat function (Figure 5). While effective, the
 250 resulting weight curve is quantized in units of
 251 the tophat’s width. To eliminate the quantiza-
 252 tion while still approximating the local density,
 253 NSClean convolves columns of the background
 254 mask with a Gaussian kernel. In the current
 255 release, the kernel’s standard deviation is hard

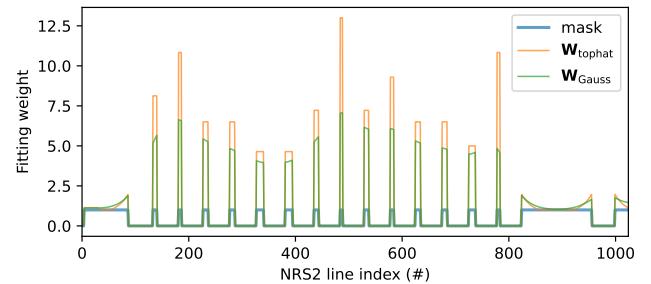


Figure 5. This figure shows the background mask and diagonal of $\mathbf{W}^{1/2}$ along the same vertical cut through NRS2 that is shown in Figure 4a. For clarity, we show only the first 1024 rows. Mask values =1 are treated as background and mask values =0 are treated as potentially illuminated. The orange curve shows the weights that result from convolving a 65 pixel wide tophat. The green curve shows that weights derived from convolving a Gaussian kernel with $\sigma = 32$ pixels. As described in the text, NSClean uses Gauss-convolution because the resulting weights are more uniform and the weight curve is not quantized.

256 coded to be $\sigma = 32$ pixels. Going forward, it
 257 may be possible to come up with something
 258 more elegant. 32 pixels seems to work well for
 259 many IFU observations.

3.4. Making Masks

This section describes how we made the masks that are shown in Figure TBD.

4. IMPLEMENTATION

NSClean is written in python-3. We chose python for compatibility with the rest of the JWST pipeline. The current NSClean version is not computationally demanding. Teams XX, YY, and the JWST Early Release Science (ERS) team TEMPLATES (Rigby et al. 2023, in prep.) have tested NSClean on typical scientific workstations and laptops and report that it works well. The typical cleaning time for one 2048×2048 NIRSpec image is a few seconds. This assumes that multithreading is turned on for the python linear algebra libraries as described in Section 4.2.

The current NSClean version works column-by-column. Since there are only 2048 pixels per column, this means that it requires very little RAM, and the time penalty for projecting out Fourier vectors using Equation 6 is small compared to using the FFT algorithm.²

4.1. Computing Requirements

When used in the recommended mask mode, NSClean is not computationally demanding. The execution time on our development server is about 6 seconds for one 2048×2048 pixel NIRSpec image. The server, which is a few years old, has $8 \times$ Intel Xeon cores running at 3.5 GHz and 250 GB of RAM. In practice, NSClean uses only a tiny fraction of the RAM. Although our server has an NVIDIA Quadro M4000 GPU with 8 GB of RAM, in practice we found that NSClean’s execution time was about the same in CPUs as in the GPU. This is because Equation 6’s matrices are not large when images are processed in columns.

² The FFT only works for fully sampled data, which we do not have.

We have also tested NSClean on a 2019 MacBook Pro. Execution time on the MacBook is about 12 seconds per image. The MacBook has an 8-Core Intel i9 CPU running at 2.3 GHz and 32 GB of RAM. Again, NSClean did not use much of this RAM. According to the Apple ActivityMonitor App, peak usage was about 150 MB.

The NSClean prototype (the NSClean1 class in the distribution) was computationally intensive. In general, we find that mask mode (the recommended NSClean class) provides better correction and is much less taxing. We have nevertheless left NSClean1 in the distribution in case anybody finds it useful. For NSClean1, using a GPU can provide a roughly a $>10 \times$ speedup compared to CPUs. Using a GPU, NSClean1’s execution time is about 3 seconds. The execution time using the server’s CPUs was a minute or two.

Our development server had the following software; Oracle Linux Server release 8.7, python-3.10.8, astropy-5.0.4, cupy-11.5.0, numpy-1.22.3, and pillow-9.3.0.

4.2. Multithreading

NSClean is not explicitly multithreaded. In practice, however, we always have multithreading turned on for python’s linear algebra libraries. As a result, when we run NSClean, it usually shows all CPUs being used because most of the work is linear algebra.

On our Intel-based computers, this is done by installing the Intel version of numpy and setting an environment variable. For our 8-core server, the python code is as follows.

```
import os
os.environ['MKL_NUM_THREADS']='8'
```

Our understanding is that on non-Intel computers, similar functionality exists, although the environment variables are different.

When a GPU is used, python’s cupy package automatically parallelizes the linear algebra operations over however many GPU cores

are available. Our NVIDIA Quadro M4000 has 1664 CUDA cores. Individual CUDA cores are slow compared to the server’s CPUs. However, because there are so many of them, they enable a $> 10\times$ speedup for NSClean1.

4.3. Installing NSClean

NSClean is a standard pip-installable python package. It is available from the NASA JWST website (NASA JWST website 2023). To install it on MacOS or Linux, change into a directory that is in your python path, and download the distribution. Then, use pip to install it,

```
pip install -e nsclean.
```

This will install nsclean as an editable package in your python path.

5. SUMMARY

Many JWST observers are finding that there is faint vertical banding and a picture frame pattern in pipeline calibrated NIRSpec images. The effect is particularly challenging for IFU observations because it can add spectral fea-

tures that are not real. This article describes the NSClean python package that uses dark areas of NIRSpec scenes to remove this noise. To use NSClean, the astronomer must provide a mask specifying which pixels are to be treated as background. For each count rate image, NSClean then: (1) computes the Fourier transform of the background using an algorithm that can handle missing data, (2) applies a low-pass filter to reduce noise, and (3) inverts the Fourier transform yielding a background model. When the background model is subtracted from the image, it removes most of the correlated noise. NSClean is simple and computationally undemanding. The NSClean python package is freely available for download from the NASA JWST Website (NASA JWST website 2023).

This work was supported by NASA as part of the JWST.

Facilities: JWST(NIRSpec)

Software: astropy (Astropy Collaboration et al. 2013, 2018)

REFERENCES

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: [10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068)
- Astropy Collaboration, Price-Whelan, A., Sip\Hocz, B., et al. 2018, AJ, 156, 123, doi: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f)
- Cooley, J. W., & Tukey, J. W. 1965, Math. Comput., 19, 297, doi: [10.2307/2003354](https://doi.org/10.2307/2003354)
- Jakobsen, P., Ferruit, P., Alves de Oliveira, C., et al. 2022, A&A, 661, A80, doi: [10.1051/0004-6361/202142663](https://doi.org/10.1051/0004-6361/202142663)
- JWST User Documentation website. 2016. <https://jwst-docs.stsci.edu/jwst-near-infrared-spectrograph/nirspec-observing-strategies>
- Loose, M., Farris, M. C., Garnett, J. D., Hall, D. N. B., & Kozlowski, L. J. 2003, Proc SPIE, 4850, 867
- NASA JWST website. 2023. <https://webb.nasa.gov/content/forScientists/publications.html>
- Rauscher, B. J. 2015, PASP, 127, 1144, doi: [10.1086/684082](https://doi.org/10.1086/684082)
- Rauscher, B. J., Arendt, R. G., Fixsen, D. J., et al. 2013, in Proc SPIE, ed. H. A. MacEwen & J. B. Breckinridge, Vol. 8860, International Society for Optics and Photonics (SPIE), 886005, doi: [10.1117/12.2025053](https://doi.org/10.1117/12.2025053)
- Rauscher, B. J., Arendt, R. G., Fixsen, D. J., et al. 2017, PASP, 129, 105003, doi: [10.1088/1538-3873/aa83fd](https://doi.org/10.1088/1538-3873/aa83fd)