DRAFT VERSION SEPTEMBER 22, 2016
Preprint typeset using LATEX style AASTeX6 v. 1.0

TESTS ON IMCOM: GENERATING NIRCAM MOSAICS

SARA CRANDALL AND BRANT ROBERTSON
Department of Astronomy & Astrophysics, University of California Santa Cruz
211 Interdisciplinary Sciences Building
Santa Cruz, CA 95064, USA

This document logs the steps taken to generate mosaic mock images for NIRCam use.

1. AN INTRODUCTION TO IMCOM

IMage COMbination (IMCOM) may be a useful tool for generating mosaic mock images for NIRCam. This software was designed by Rowe et al. (2011) for use on WFIRST, and it allows undersampled images to be combined into an oversampled image. This would be useful for NIRCam, as some bands provide undersampled images. DRIZZLE is commonly used for image combination, but may have resolution loss resulting from reduced control of the PSF in the combined image. The Fourier/linear algebra technique is also useful, but can be less reliable when positions are not well known. A least-squares tool is beneficial in that output noise can be minimized. IMCOM uses a combination of these three methods to reconstruct an image with a controlled PSF, knowing the positions of exposures, that also minimizes noise.

To ensure that IMCOM is the right tool for NIRCam use, we will run a few tests. As described in the software's documentation, a baseline set of arguments are required. The command line looks like

$$./\mathrm{imcom}\ \ <\mathrm{config\_file}> <\mathrm{U/S}> <\mathrm{U/S\_max}> <\mathrm{U/S\_tol}>. \tag{1}$$

Here, the input $<\mathrm{U/S}>$ will either minimize the leakage objective, U_α, (the local leakage or difference between the desired PSF and the reconstruction) or the noise covariance for each output pixel, $\Sigma_{\alpha\alpha}$. $<\mathrm{U/S\_max}>$ defines the maximum leakage objective or noise covariance, $U_\alpha^{\max}$ and $\Sigma_{\alpha\alpha}^{\max}$ respectively, depending on the previous option. $<\mathrm{U/S\_tol}>$ defines $\Delta U_\alpha^{\mathrm{tol}}$ or $\Delta\Sigma_{\alpha\alpha}^{\mathrm{tol}}$ that specify when the iteration should stop. The program will finish when $U_\alpha^{\max} - \Delta U_\alpha^{\max} < U_\alpha \leq U_\alpha^{\max}$ or $\Sigma_{\alpha\alpha}^{\max} - \Delta\Sigma_{\alpha\alpha}^{\max} < \Sigma_{\alpha\alpha} \leq \Sigma_{\alpha\alpha}^{\max}$.

2. TEST_0

Test_0 will determine which IMCOM inputs are useful for our purposes by varying the above options.[1] IMCOM has two examples included with the package (userxy0 and userxy1) which we will use those for our initial tests. Userxy1 gives the user more freedom to designate the input pixel locations.

To begin, example userxy0 will be used and the U/S option will be varied. The following inputs were used:

$$\mathrm{config\_example\ U\ 1.d} - 3\ 1.\mathrm{d} - 6^{2} \tag{2}$$

and

$$\mathrm{config\_example\ S\ 1.d} - 3\ 1.\mathrm{d} - 6. \tag{3}$$

The results are seen in Figure 1, and there does not appear to be a significant visual difference between the two options.

Next, the userxy0/userxy1 option will be varied. The input

$$\mathrm{config\_example\ U\ 1.d} - 3\ 1.\mathrm{d} - 6 \tag{4}$$

is used. Figure 2 shows the results. The userxy1 option is less defined, which may not be surprising as the userxy1 option allows a user input for pixel centering and so may be a more accurate image.

sacranda@ucsc.edu, brant@ucsc.edu

[1] There are other inputs that can be specified such as the convolution and objective kernels, designated as "soft inputs" in Rowe et al. (2011), but they won't be varied at the moment.

[2] Double precision floating point numbers are used

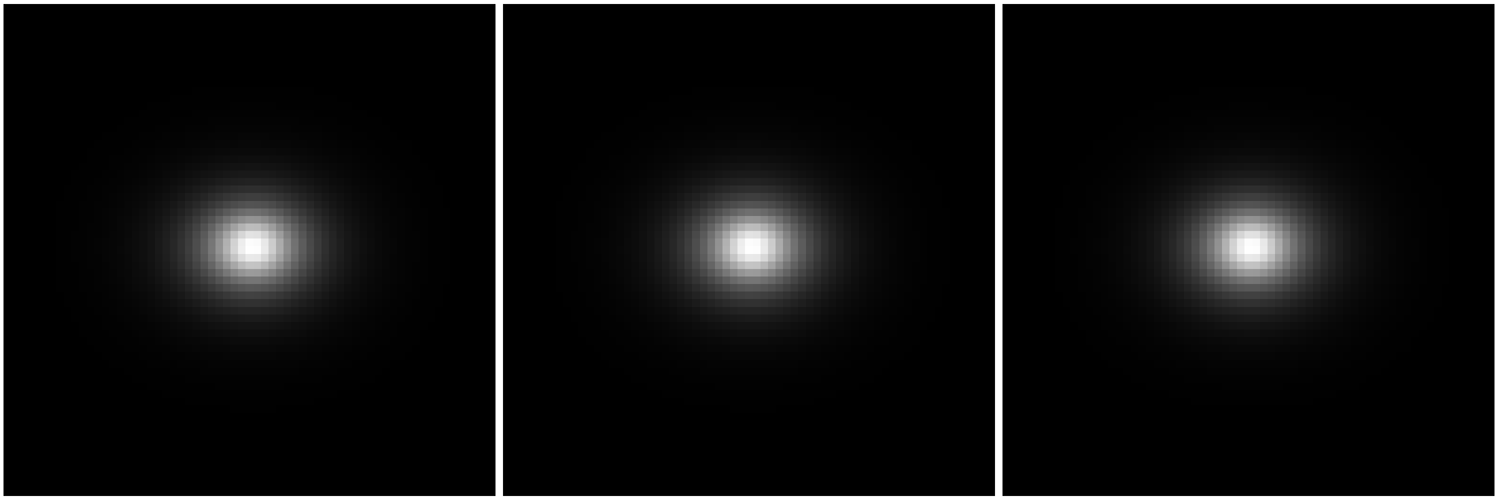


Figure 1. The leftmost and center images use the "U" and "S" options, respectively. The rightmost image is the ideal output included with IMCOM.

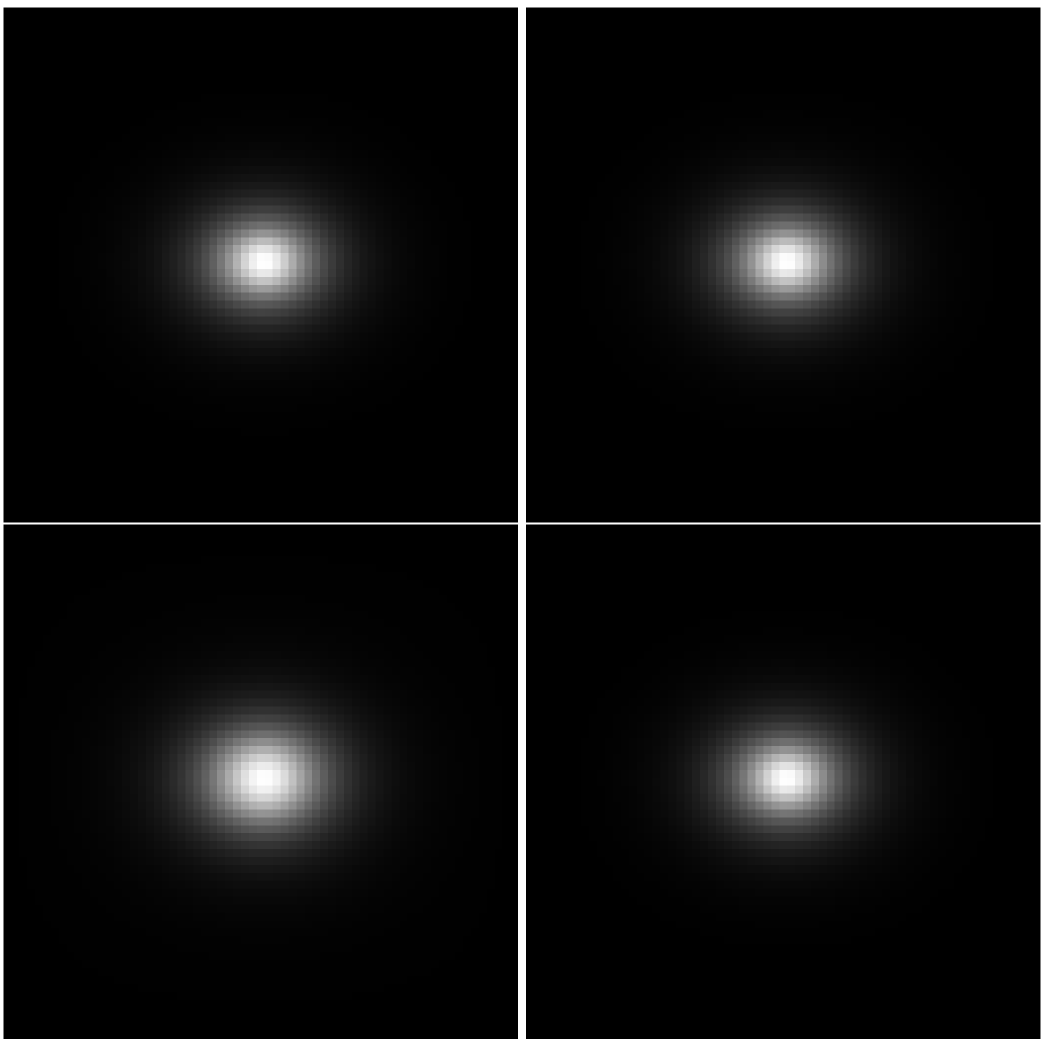


Figure 2. The top (bottom) images uses the "userxy0" ("userxy1") option. The left (right) panel is the output (ideal) image.

Finally, the U_max and U_tol options will be varied. To begin, the userxy0 example will be used while minimizing the leakage objective ("U" option). The following inputs are shown in Figure 3:

$$\text{config\_example U } 1.\text{d}-1\ 1.\text{d}-6 \tag{5}$$

and

$$\text{config\_example U } 1.\text{d}-6\ 1.\text{d}-6. \tag{6}$$

There are no noticeable differences in the images here. More U_max and U_tol values should be tested to find ideal

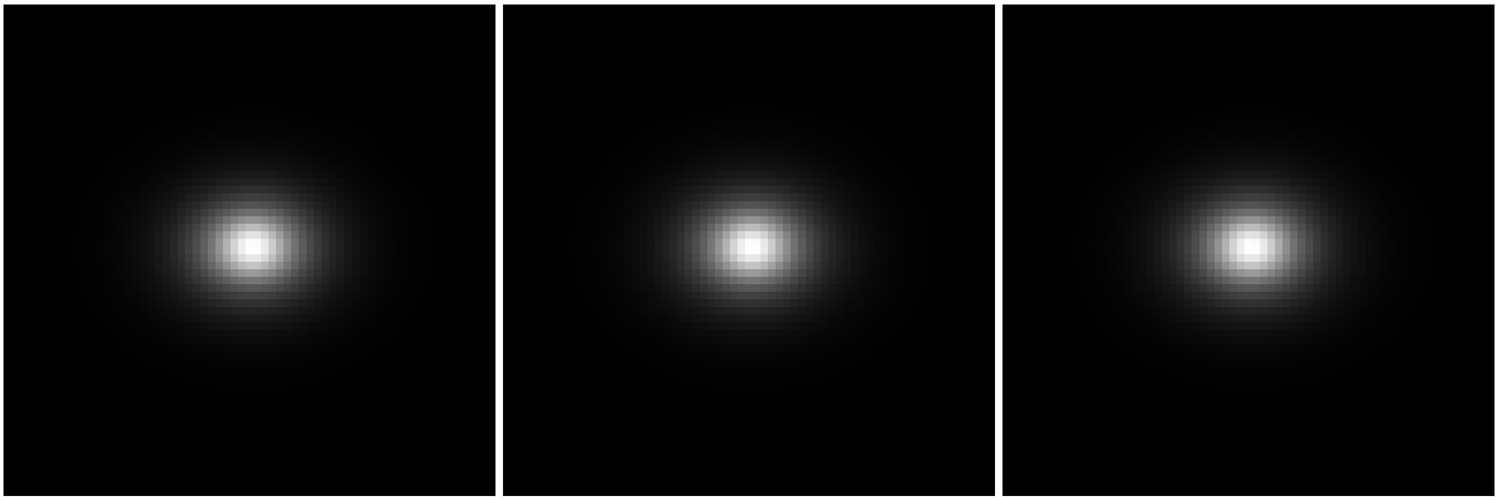


Figure 3. The leftmost image reflects a U_max $= 1.\text{d}-1$ and U_tol $= 1.\text{d}-6$ (userxy0 and U options). The center image has U_max $= 1.\text{d}-6$ and U_tol $= 1.\text{d}-6$ inputs. The Rightmost image is an ideal output.

(and realistic) images.

3. TEST_1

We will move forward with using the userxy1 example, as the pixel locations need to be provided. We will also focus on minimizing the leakage objective, as we are concerned with how well the output image matches the inputs.[3] With these options we will continue varying U_max and U_tol, and call this exercise "Test_1." Figure 4 shows several examples of this test. U_max and U_tol were varied from 1.d-1 to 1.d-10, as IMCOM gives warning messages for smaller values.

It will be useful to quantify the images in Figure 4. We do so by finding the L1 error norm for each output given by

$$\text{L1} = \sum_i |H_i - I_i|, \tag{7}$$

where H_i and I_i are the pixels for the output image and ideal output image respectively. A color plot of this error norm is shown in Figure 5. This figure shows that, ideally, we should use inputs that are small, preferably less than 1×10^{-6} for both U_max and U_tol.

4. TEST_2

We now move on to "Test_2," where we see how well IMCOM can handle noise in the input images. First, we will use small U_max and U_tol inputs to decrease the differences in output and ideal output images (as deomonstrated in Test_1). We use the IMCOM input

$$\text{config\_example U } 1.\text{d}-9\ 1.\text{d}-10 \tag{8}$$

for the userxy1 example. The overall goal here is to investigate how the L1 error norm is affected by the signal to noise ratio (S/N).

[3] Minimizing U_tol can result in a reconstruction that is not trustworthy, as the algorithim may not reach an ideal reconstruction.

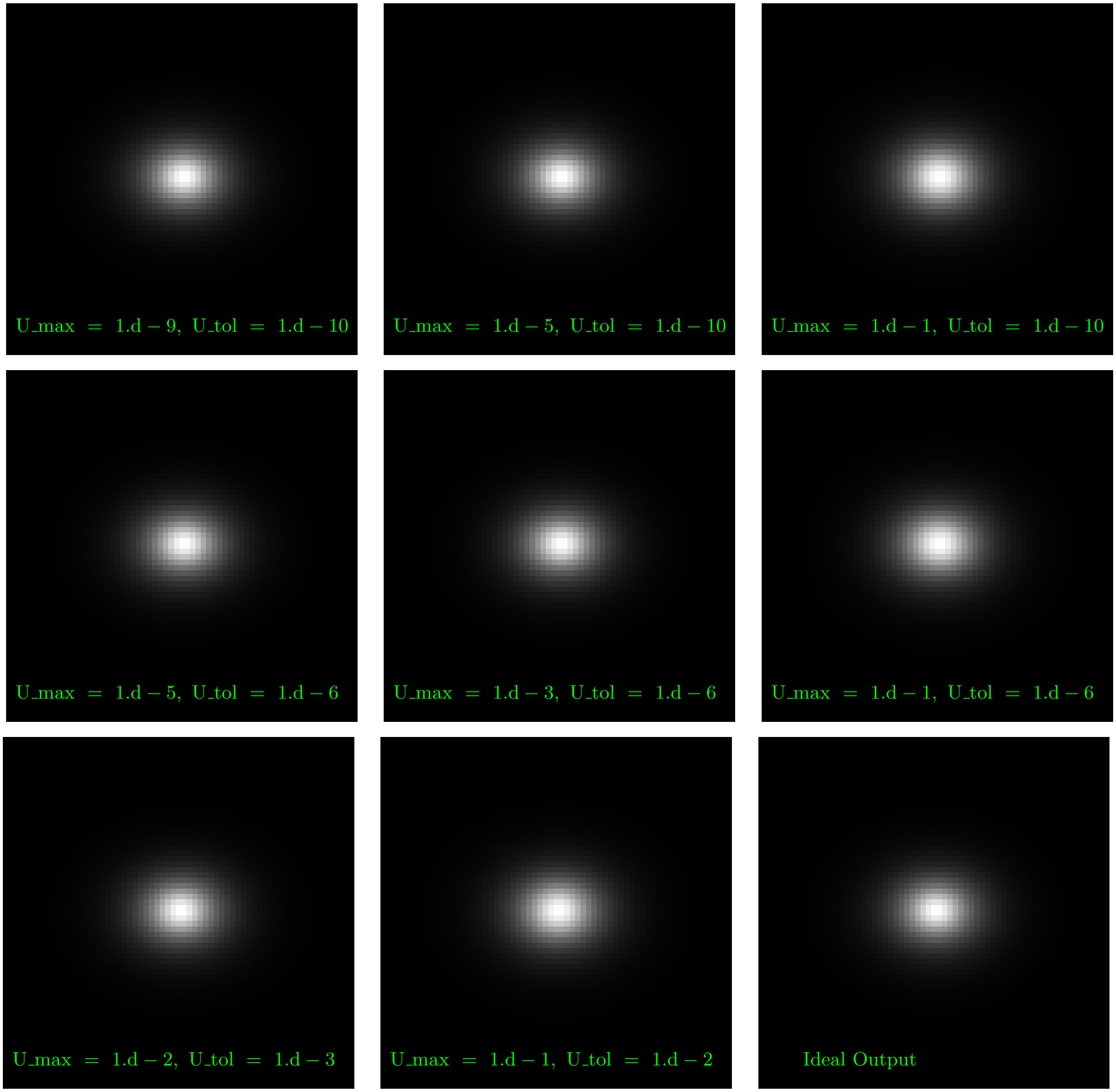


Figure 4. Output images using userxy1 example, U option, and varying U_max & U_tol.

To add noise, we draw random values from a Poisson distribution given by

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{9}$$

and add to existing input images. Here λ depends on the S/N, which we will vary. That is

$$\lambda = \left(\frac{< S_p >}{S/N}\right)^2, \tag{10}$$

where $< S_p >$ is the average signal from the input images given by

$$< S_p > = \frac{\sum_i p_i \Delta A_i}{\sum_i \Delta A_i}. \tag{11}$$

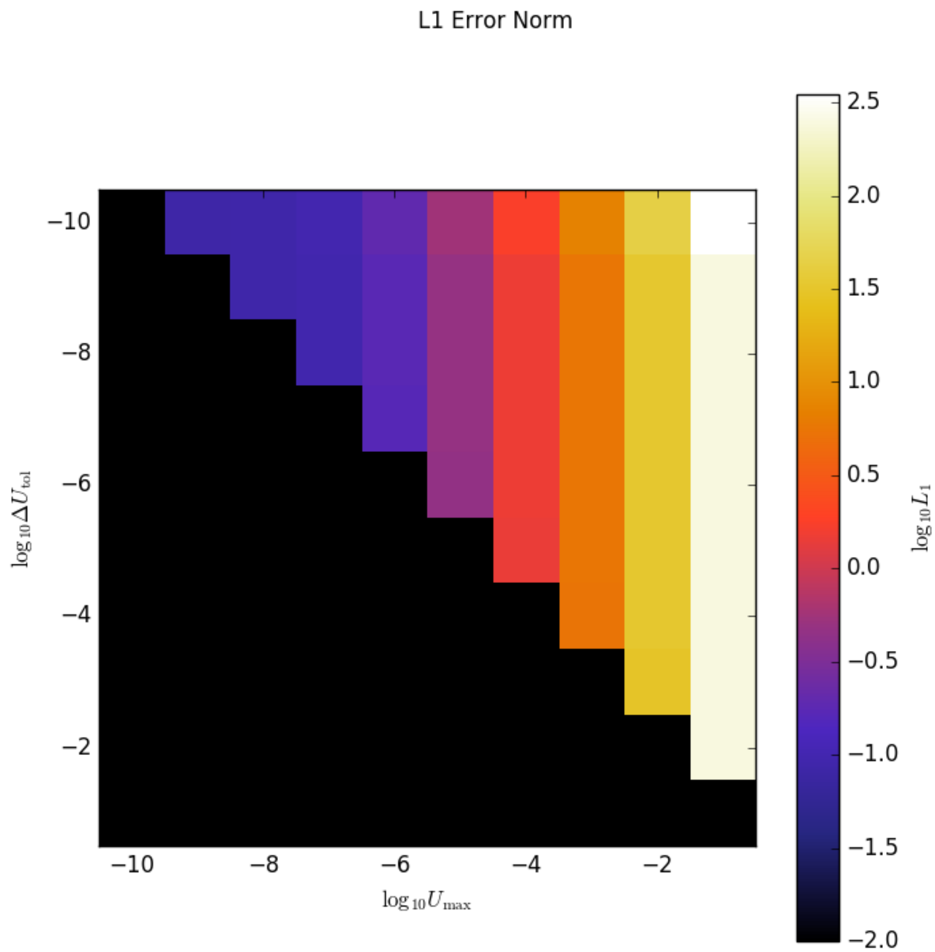


Figure 5. Log color plot of L1 error norms.

Here, p_i is the pixel and ΔA_i is the area of the pixel.

We generate noisy images for $S/N = 1-50$. Figure 6 shows the input, output, and ideal output images for $S/N = 5$ as an example. Rescaling shows that the pixels of the output image are being distorted by the rightmost and topmost pixels.

After examining the pixel locations for the IMCOM input configuration file, we see that the four 32×32 pixel size input images are being combined into a 65×65 output image. It is possible to cut off the last column and row of the output image to solve the output image distortion problem (that is, generating a 64×64 pixel output image). Figure 7 shows the input, output, and ideal images after generating a 64×64 reconstruction for $S/N = 5$. This is a reasonable solution, as we are allowed to control the pixel locations in the userxy1 example.

Next, we will check that we are accurately adding the noise to the input images. As a quick first test, we'll check

$$S/N = \frac{< S_p >}{< S_k >}, \tag{12}$$

where $< S_k >$ is the average noise given by

$$< S_k >= \frac{\sum_i k_i \Delta A_i}{\sum_i \Delta A_i}. \tag{13}$$

Here k_i is the noise pixel.

Table 1 shows that we get close to what is the expected S/N for each exposure, however, the discrepancy gets larger as S/N increases. As we will not be concerned with $S/N > 15$, this is reasonable.

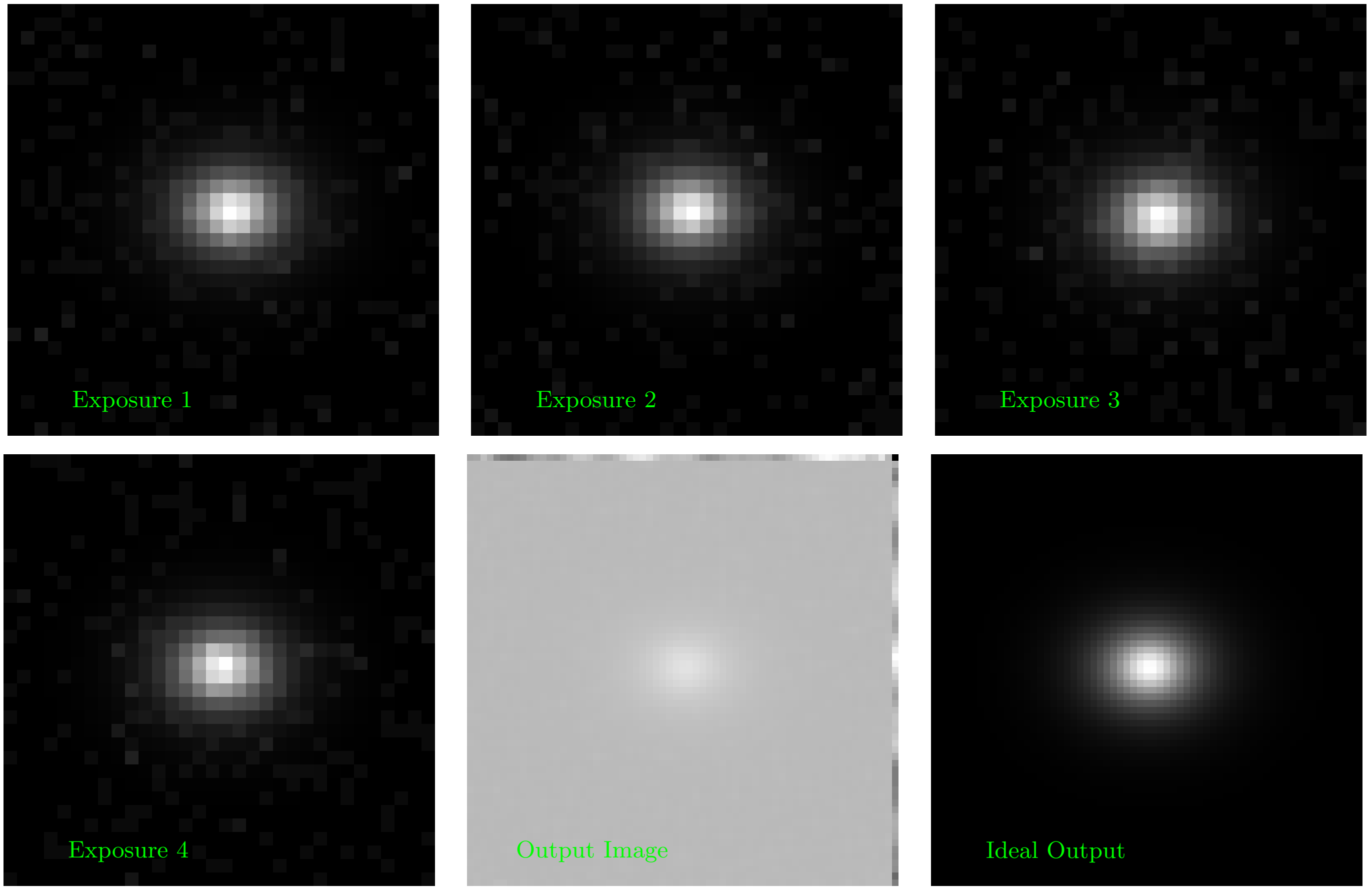


Figure 6. Noise added to input images for the $S/N = 5$ case (65×65 reconstructed image).

Table 1. First S/N test

S/N	Exposure	$\frac{<S_p>}{<S_k>}$
5	1	5.46253286315
5	2	4.94753975121
5	3	4.99443967426
5	4	5.40672406067
Avg:		5.2028090873225
17	1	22.3150278665
17	2	19.0705168592
17	3	18.083316062
17	4	15.2015140257
Avg:		18.66759370335
43	1	65.5503943578
43	2	49.9465917741
43	3	55.2017016628
43	4	41.9561787108
Avg:		53.163716626375

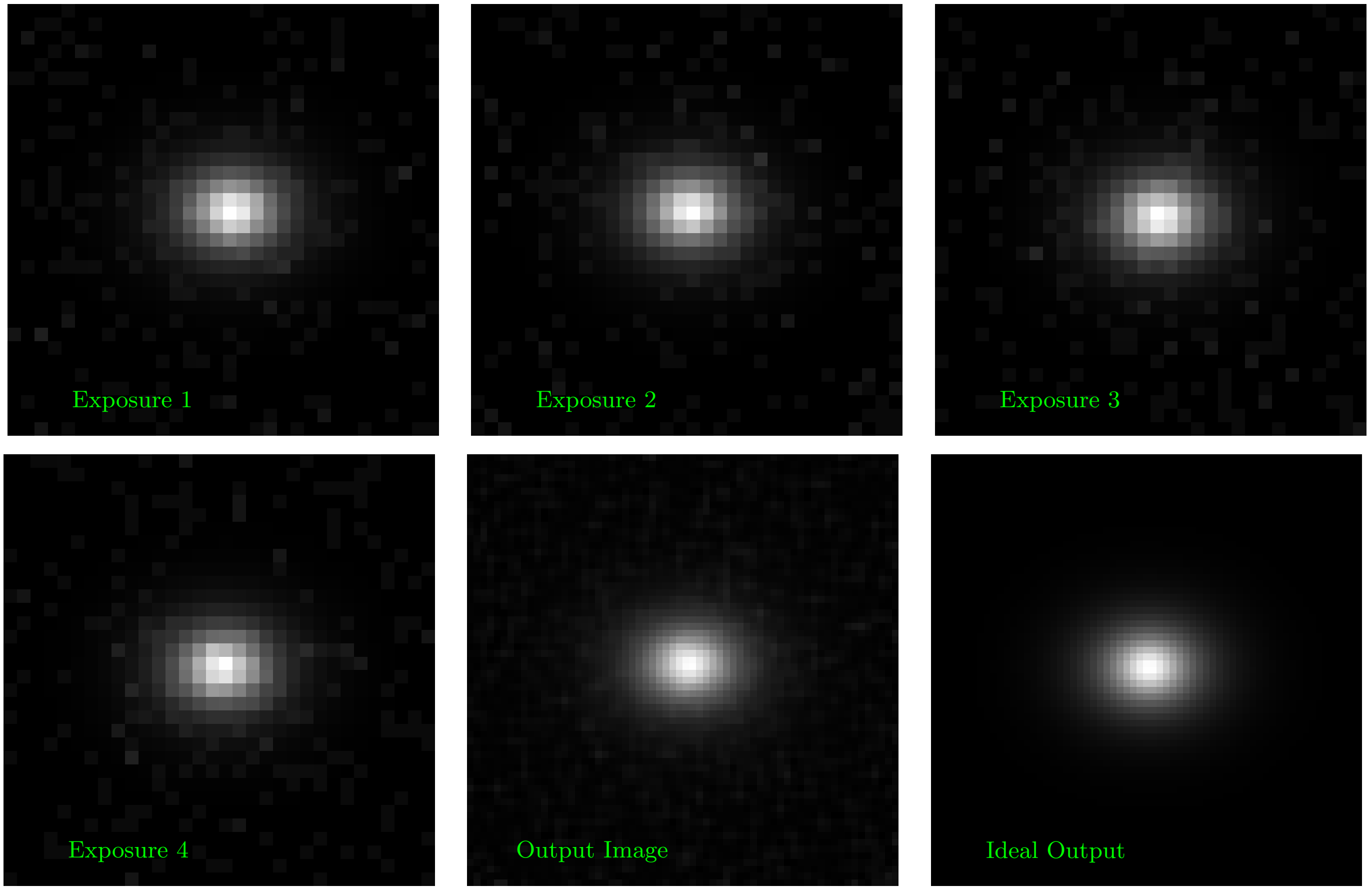


Figure 7. Noise added to input images for the $S/N = 5$ case (64×64 reconstructed image).

Next we compare the probabilities of the outputs from the random Poisson distribution and compare to the theoretical for a given S/N. Table 2 gives the probabilities for the $S/N = 5$ and $S/N = 40$ cases. The observed and expected probabilities, P(k), are in good accordance, and we are satisfied with the accuracy of the noise that is added to the input images.

Moving forward, we can plot the L1 error norm as a function of S/N. Figure 8 shows this relationship. The right panel shows a power law fit.

The large L1 values may be an indication that we are not getting a good representation of the noise from the input images. To check, we will pull noise from the inner portion of the input images that account for 90% of the pixels. This corresponds to the inner 16×16 square of the 32×32 input images. Figure 9 shows the new noisy images using this technique. This generated noise visually appears more reasonable. We can also plot L1 vs. S/N (see Figure 10) to quantify this, and indeed, L1 has decreased for larger S/N compared to using the entire image to generate noise.

To be more practical, we're going to focus on $0 < S/N < 15$. A plot of L1 vs. S/N for more values between this range is shown in Figure 11.

To further check how the noise is being handled by IMCOM, we can plot χ^2 vs. S/N (see Figure 12). We define χ^2 as

$$\chi^2 = \frac{1}{N-1} \sum_i^N \frac{(H_i - I_i)^2}{|H_i|}, \tag{14}$$

where N is the total number of pixels, and H & I are the output and ideal image for pixel $i = 1, 2, ..., N$. This plot is promising, as χ^2 is reasonable for values $S/N > 2$. There are a couple of deviations from the general trend around $7 < S/N < 8$, but it is within normal expected error.[4]

[4] We investigated this further by comparing observed and expected probabilities from the Poisson distributed noise values (similar to Table 2), and found the differences to be within reason. We also used different seeds for our randomly pulled values, and found a similar bump in the L1 vs. S/N plot.

Table 2. Second S/N test

k (Event)	# of Events[a]	Observed Probability[b]	P(k)[c]	# of Events[a]	Observed Probability[b]	P(k)[c]
		$S/N = 5$			$S/N = 40$	
Exposure 1						
0	851	0.8310546875	0.814773610898	1008	0.984375	0.974719420938
1	155	0.1513671875	0.166902285958	15	0.0146484375	0.0249582978245
2	16.	0.015625	0.0170945479122	1	0.0009765625	0.000319536379862
3	2	0.001953125	0.00116724412189	0	0.0	2.72730933745e-06
4	0	0.0	5.97760253917e-05	N/A	N/A	N/A
Exposure 2						
0	841	0.8212890625	0.814762134521	1000	0.9765625	0.974717704769
1	153	0.1494140625	0.166911411376	24	0.0234375	0.0249599700487
2	29	0.0283203125	0.0170966580718	0	0.0	0.000319579762318
3	1	0.0009765625	0.00116746847849	0	0.0	2.72786717515e-06
4	0	0.0	5.97916260543e-05	N/A	N/A	N/A
Exposure 3						
0	836	0.81640625	0.814769469906	1004	0.98046875	0.974718801699
1	166	0.162109375	0.166905578678	18	0.017578125	0.0249589012069
2	21	0.0205078125	0.0170953093009	2	0.001953125	0.000319552033043
3	1	0.0009765625	0.00116732507249	0	0.0	2.72751061102e-06
4	0	0.0	5.9781654175e-05	N/A	N/A	N/A
Exposure 4						
0	848	0.828125	0.814757990621	999	0.9755859375	0.974717085087
1	157	0.1533203125	0.16691470635	24	0.0234375	0.0249605738613
2	18	0.017578125	0.0170974200418	1	0.0009765625	0.00031959542775
3	1	0.0009765625	0.00116754949669	0	0.0	2.72806862017e-06
4	0	0.0	5.97972599327e-05	N/A	N/A	N/A

[a] The number of times k occurs from the randomly drawn Poisson distribution.
[b] The observed probability for each k.
[c] The theoretical probability from Equation 9.

5. TEST CONCLUSIONS

Because IMCOM (Rowe et al. 2011) makes use of the benefits from common mosaicing tools, it is a ideal software package to generate JWST NIRCam mock mosaic images. It will be useful in that it gives control of the PSF in the combined image, it minimizes noise, and is used by knowing the positions of the pixels.

After running several tests on provided example images, it appears that IMCOM allows enough input freedom for the user. As an initial test, IMCOM can handle added noise to input images for at least $S/N > 2$.

In conclusion, IMCOM is a promising tool for NIRCam use, and we will continue to use it by testing mock images generated by Christopher Willmer.

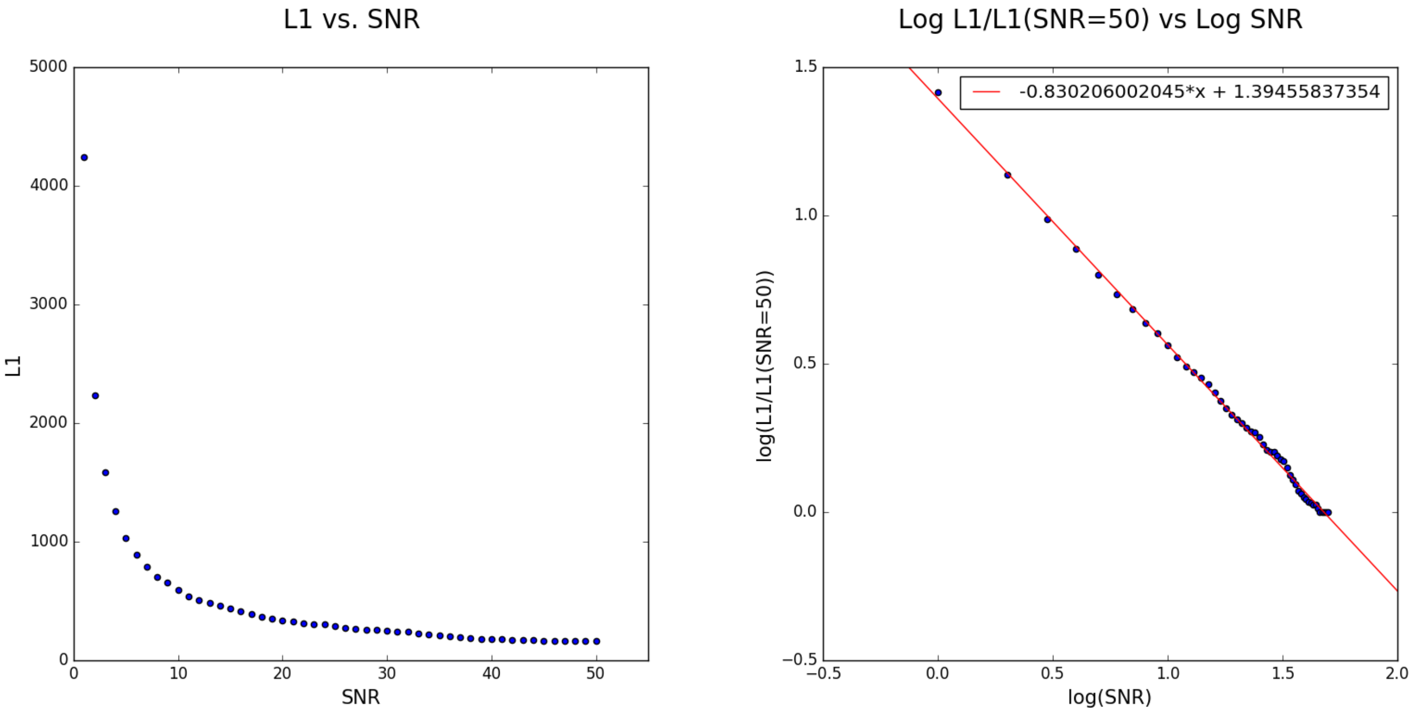


Figure 8. The left plot is the L1 error norm as a function of S/N. The right plot shows the linear fit (red line) to a log plot of L1/L1($S/N_{\rm max}$) vs. S/N. It appears to be a power law.

REFERENCES

Rowe, B., Hirata, C., & Rhodes, J. 2011, ApJ, 741, 46

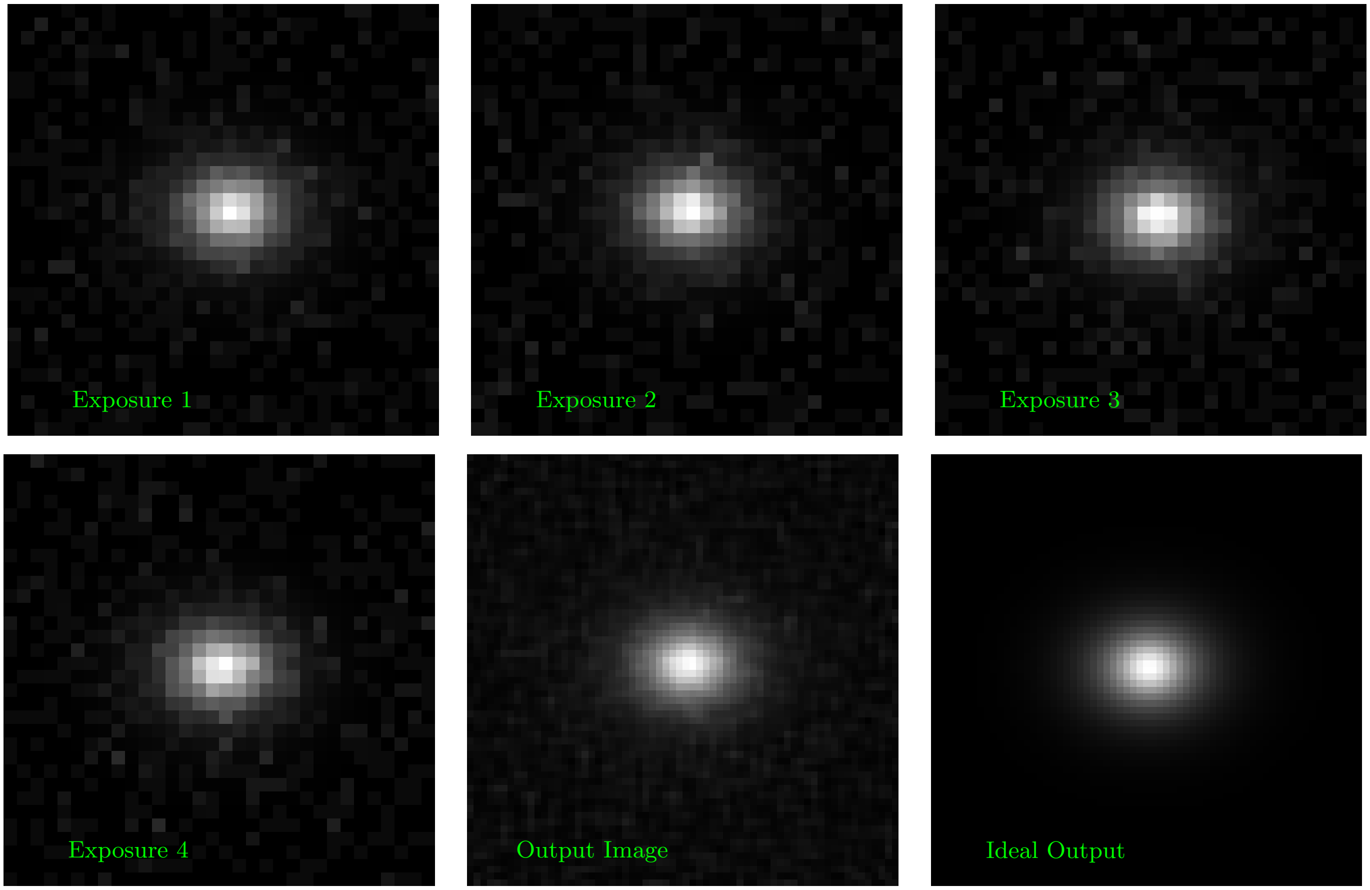
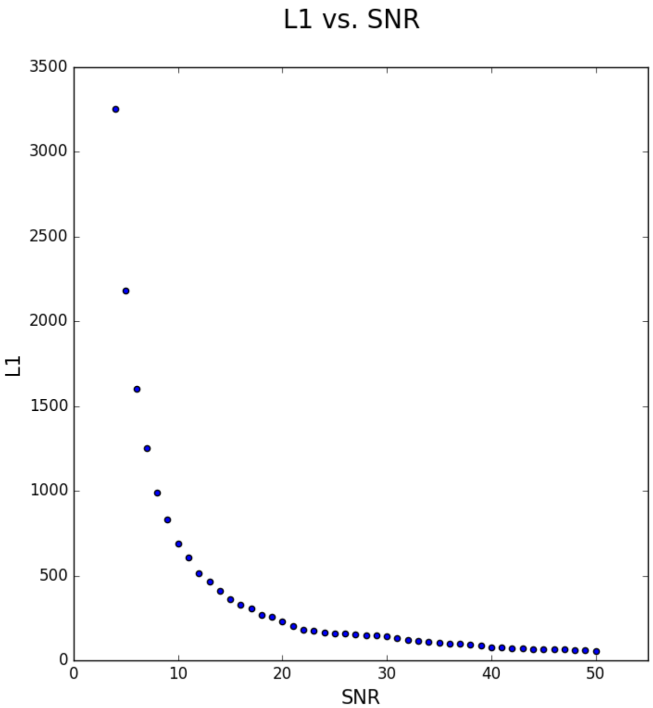


Figure 9. Noise added to input images using inner 90% of pixels for the $S/N = 5$ case.



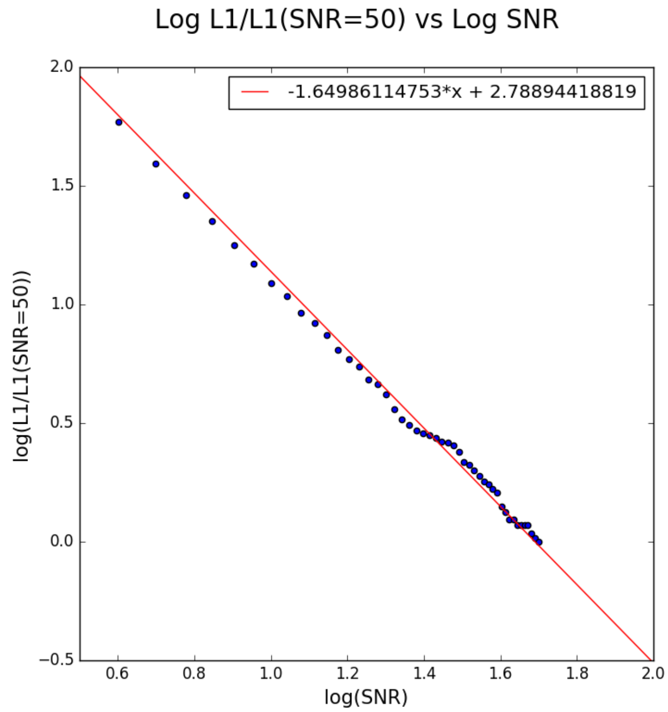


Figure 10. The left plot is the L1 error norm as a function of S/N. The right plot shows the linear fit (red line) to a log plot of L1/L1($S/N_{\rm max}$) vs. S/N) It appears to be a power law.

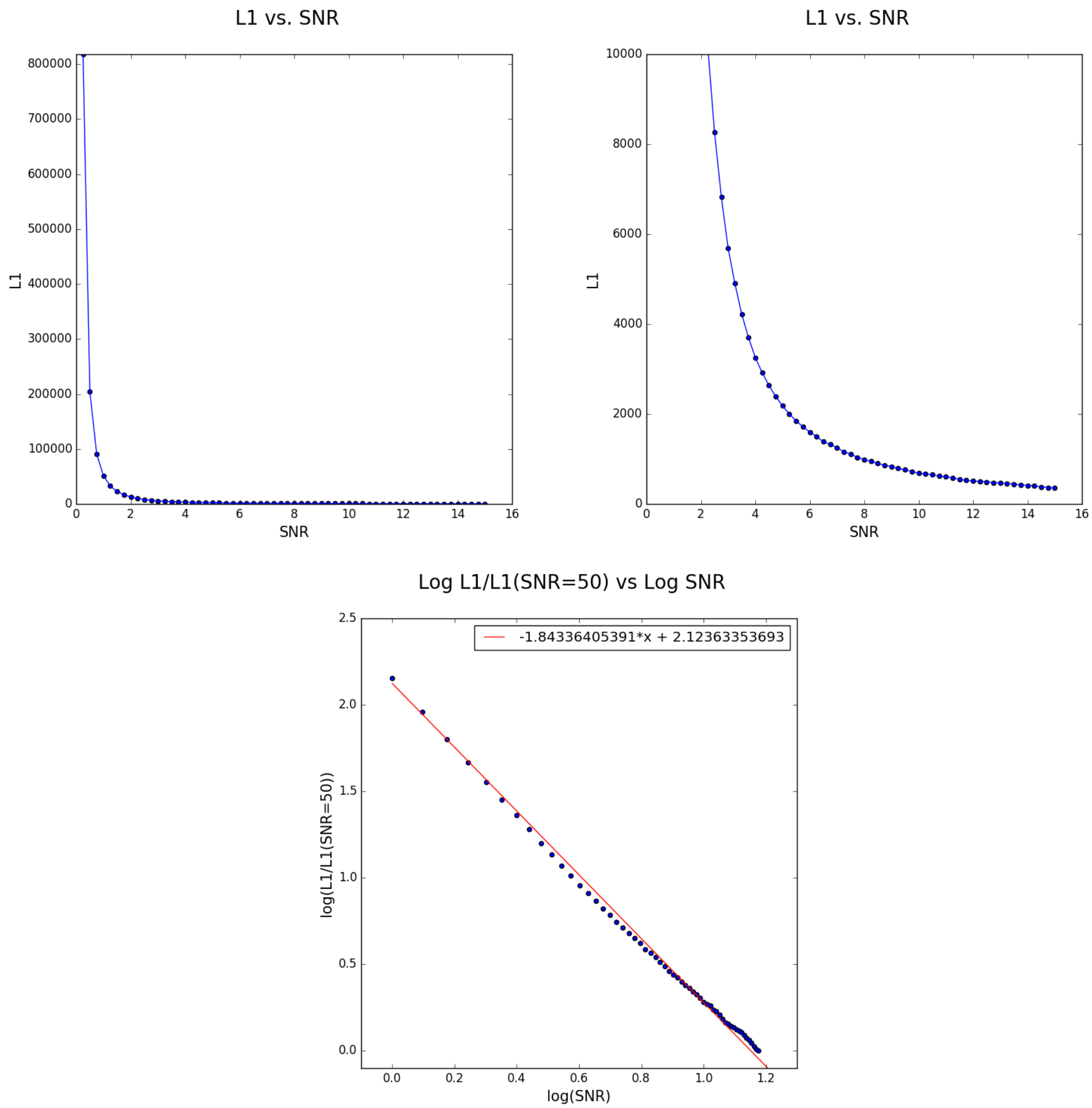


Figure 11. Focusing on values with $S/N < 15$; the top two plots show the L1 error norm as a function of S/N, with the right plot zoomed in. The bottom plot shows the linear fit (red line) to a log plot of $\mathrm{L1}/\mathrm{L1}(S/N_{\mathrm{max}})$ vs. S/N).

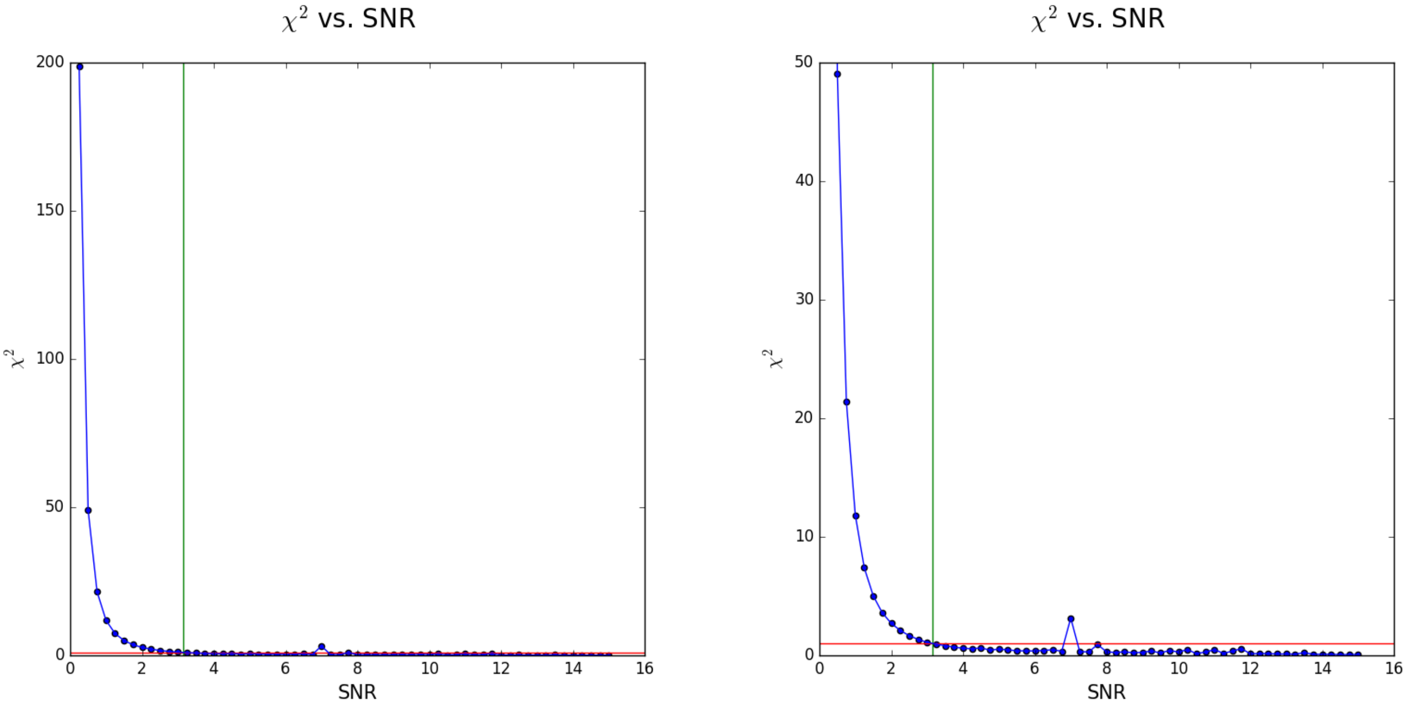


Figure 12. χ^2 vs. S/N. for $S/N < 15$. The right plot is a zoomed in version. The red line shows $\chi^2 = 1$ and the green line shows where χ^2 intersects with 1.