Draft version January 31, 2019
Typeset using LaTeX **preprint** style in AASTeX62

# Simulating NIRCam images with *Guitarra*

Christopher N. A. Willmer,[1] Karl Misselt,[1] Sandro Tacchella,[2] Daniel J. Eisenstein,[2] Kevin Hainline,[1] Benjamin Johnson,[2] Jarron Leisenring,[1] Marcia J. Rieke,[1] and Christina C. Williams[1]

[1]*Steward Observatory, University of Arizona*
*933 North Cherry Avenue*
*Tucson, AZ 85721, USA*
[2]*Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA, 02138, USA*

## ABSTRACT

*Guitarra* is an image simulator that makes NIRCam mock scenes through the combination of an input catalog, the JWST pointing position and NIRCam engineering data. *Guitarra* was developed to test the object detection, photometry and classification algorithms, quantify the selection functions and overall preparation of the GTO observations of the joint NIRCam and NIRSpec extragalactic teams.

*Keywords:* catalogs — surveys

## 1. INTRODUCTION

*Guitarra*[1] is collection of *FORTRAN* code with *perl* and *python* wrapper scripts that reads catalogs of sources with positions, shapes and magnitudes to generate a "scene" that provides a visual counterpart to the catalog. There are several motivations to develop *Guitarra* – test the ability of finding very faint galaxies, developing techniques that will optimize their photometry and classification, test the efficacy of the mosaic-making algorithms. All of these are essential to understand and quantify the selection effects at play in the observed samples and how to correct them.

*Guitarra* was designed specifically to address the needs of the NIRCam Extragalactic survey, as at the time coding was initiated ($\sim$ April 2011) no other simulator existed. Since then other simulators that can generate NIRCam images have been written or are under development. Of note are pyNRC[2] by Jarron Leisenring, which can generate scenes using the full field or the NIRCam coronagraph for direct imaging or slitless spectroscopy; the Space Telescope Image Product Simulator (STIPS)[3], which can generate scenes using Sersic models and adding a mix of stellar population models; mirage[4] (Hilbert et al. 2017), which also generates imaging and slitless spectroscopy scenes for NIRCam and NIRISS and PHOSIM[5] originally developed for LSST, which is being adapted for JWST+NIRCam by the original developers and Eiichi Egami of the NIRCam Team; PHOSIM differs from the other simulators in its use ofoptical modeling of JWST and NIRCam for the instrument characteristics.

Corresponding author: Christopher N. A. Willmer
cnaw@as.arizona.edu

[1] available at https://github.com/cnaw/guitarra
[2] https://github.com/JarronL/pynrc
[3] https://jwst.stsci.edu/science-planning/proposal-planning-toolbox/image-and-spectroscopy-simulator
[4] https://github.com/spacetelescope/mirage
[5] https://www.lsst.org/scientists/simulations/phosim

## 2. INPUT PARAMETERS

The outline of *Guitarra* is shown in Fig.1 where a catalog of sources, the positions of the telescope and the detector footprints are provided as input to generate an astronomical scene. To create the scenes, *Guitarra* traces each photon associated with an object, and scatters it by the Point Spread Function (PSF) and the inter-pixel capacitance (IPC). Stars can be approximated as point sources while galaxies are modeled by Sersic profiles which can contain up to four components. The photons associated to extended objects (galaxies, cosmic rays) as well as the perturbation due to the PSF are modeled using the Monte Carlo Inverse Transform Sampling method, described by Press et al. (1992). The random number generator adopted by *Guitarra* – *randgen*[6] – was written by Richard Chandler and Paul Northrop for use in the prediction of weather patterns.

To run *Guitarra* the user needs to provide some basic parameters: a catalog of sources, a catalog of dither positions, files with the detector footprints and the Point Spread Function (PSF) adequate for each filter and the full instrument sensitivity curves. These are summarized in Fig. 1. The catalog that *Guitarra* expects should contain an object (numeric) identification, the Right Ascension and Declination, a fiducial magnitude, the redshift, some shape parameters – semi-major and semi-minor axes, the position angle of the *object*, again relative to N going to E, the Sersic index and a magnitude or sequence of magnitudes in different filters. *Guitarra* can handle up to 4 distinct Sersic components, such that an object can have a stellar source at its center, and one or more $R^{\frac{1}{n}}$ components. The catalog must have a single line header where all columns are identified.

The list of dither positions should identify the NIRCam *aperture*'s [7] position on the sky (ra0, dec0), its position angle (pa) relative to North, with pa=90°being towards the East; the detector readout mode; the number of groups; the number of sub-pixel dither moves; the number of larger dither moves, source catalog name and filters being used.

These can be set by the user editing a script (the intention is to have this be input by a GUI) or these can be read from a file that is exported by APT using the *export* menu and then *visit coverage* on the pop-up window. This file provides the *aperture* being used and the positions for each dither (which includes the major *and* sub-pixel moves). This file also shows the total exposure time for the combined ensemble of moves (instead of the exposure time associated to each move – therefore some manipulation of the output needs to be carried out). In addition one has to add by hand the filter associated to each dither position, the readout mode and the number of groups.[8] As the exported file only shows the coordinates for the primary instrument, in cases where NIRCam is the parallel instrument, its coordinates on the sky need to be converted from the primary instrument's position, which is done using the Science Instrument Aperture File (SIAF) positions. In addition, when NIRCam is used in parallel, the readout modes, exposure times and filters need to be added manually[9].

The detector footprint uses the calibration images obtained during the NASA/GSFC ISIMCV3 cryogenic test. On ISIMCV3 the final ASIC tuning of all detectors was accomplished, being a necessary step to have robust estimates of several of the detector characteristics such as the measurement

[6] http://www.ucl.ac.uk/~ucakarc/work/randgen.html

[7] In the case of NIRCam the aperture can be an individual SCA (NRCA1, NRCBLONG etc.), a channel (NRCBALL) or the entire detector (NRCALL)

[8] A substantiated request has been made to the STScI Helpdesk to so the programmers add these parameters to the *exported* output of APT. Until that happens the alternative is to read the XML output from APT or add missing data by hand.

[9] As this is being written (April 2018) the file exported from APT shows incorrect positions for *NIRSpec* when it is the primary instrument (the Aladin positions are correct); STScI programmers are tracing this bug

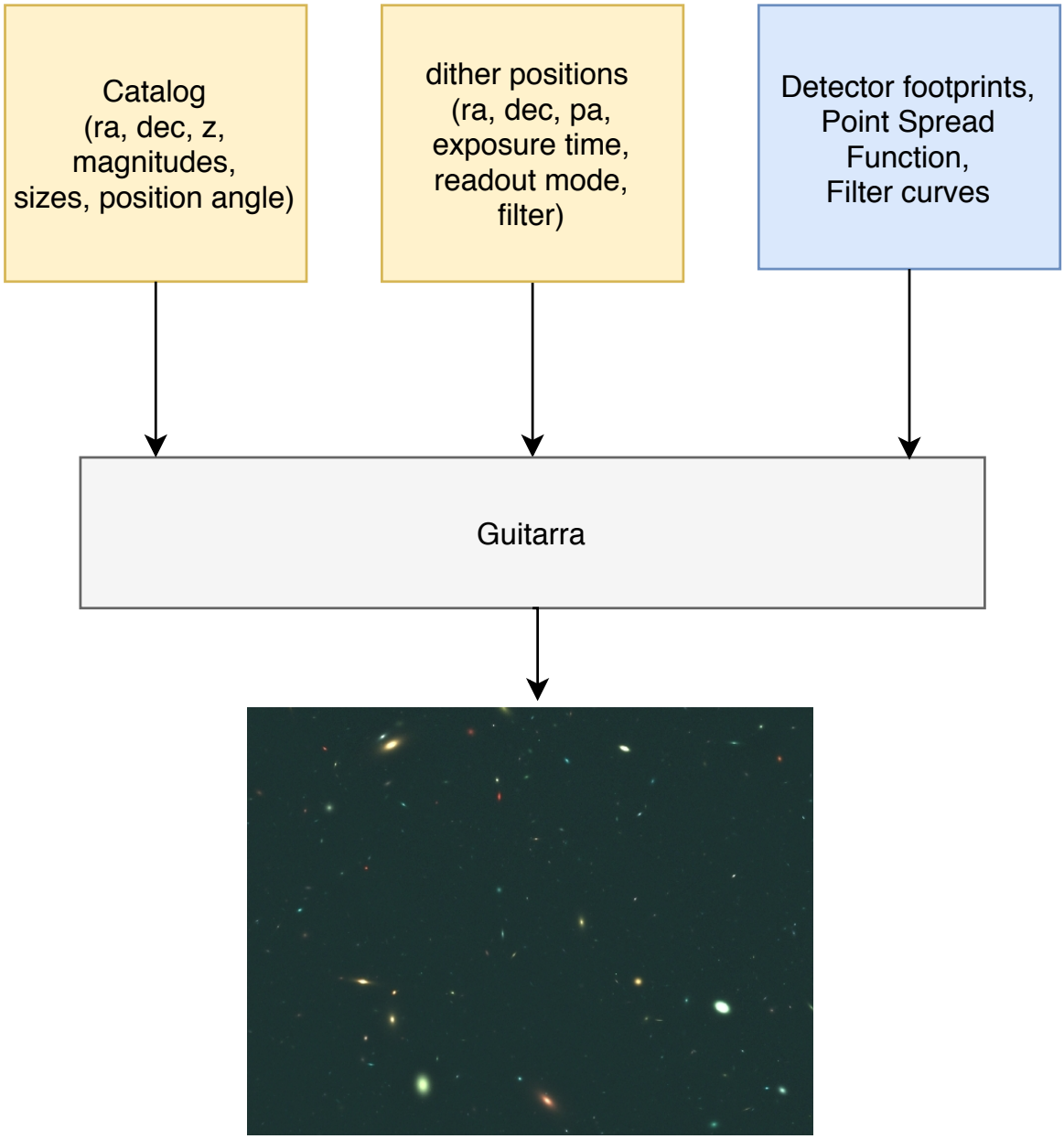


**Figure 1.** Flow chart for *Guitarra*, where data coming from an object catalog, the region on the sky being simulated and the detector characteristics are combined to produce a simulated scene. The scene being shown is a small region of a mosaic that used the *JADES* Mock Catalog of Williams et al. (2018) and the output from the APT corresponding to JWST program 1180. This combines simulations for images in the F090W, F115W and F356W filters of NIRCam each mosaiced using the *swarp* program of Bertin et al. (2002).

of the dark current, the detector bias, gain, and the linearity correction. Ground-based flatfields were also obtained for the majority of filters (exceptions being the two shortest-wavelength ones – F070W and F090W).

The calibration images used in the simulation are the bias level, which is set by the combination of the kTC noise and the ASIC tuning level; the dark current, which is estimated from linear fits to

dark images (Table 6) obtained in ramps with 108 frames, but turns out to be negligible for most of the short-wavelength detectors (and can be ignored); pixel-to-pixel gain and bad-pixel maps and look-up linearity tables that link the total charge that is incident in a pixel with that observed.

## 3. RUNNING *Guitarra*

The first step in the process is to combine the information from the source catalog, the sky position and the detectors into the *Guitarra* input file. This step is carried out by wrapper scripts.

*Guitarra* starts by reading these parameters and initializes the random number generator. The core of *Guitarra* is the *add_up_the_ramp.f* routine, which uses sequence of operations shown in Fig. 2. The type of random sampling used is different (e.g., Poisson, Gaussian) for the various steps, and these are summarized in Table 3.

The routine starts by reading the files that determine the detector footprint and initializes the array, storing the simulated detector baseline by using Poisson variates to sample the bias image. This array is stored and added to the final counts in a given group after the latter are distorted by the detector non-linearity. Other buffers are used to store counts. One is associated to individual groups, which contains the total counts in a group. Another contains the total counts in a ramp and is used to calculate the non-linearity distortion. The data cube stores the distorted counts.

The fits data cube that stores the raw scene is created by *open_big_fits_cube.f* which also sets the image size (in the third dimension) and adds the header keywords required for the reduction using the University of Arizona *ncdhas* code (Misselt 2019) and those required for the STScI reduction. The latter are listed in https://mast.stsci.edu/portal/Mashup/clients/jwkeywords/index.html which are updated a few times per year.

Galaxies are added by subroutine *add_galaxy_component.f* and are described by up to four components with Sersic profiles such that

$$m_{tot} = -2.5 Log_{10}(\sum_{i=1}^{\leq 4} flux_i). \tag{1}$$

The flux of individual components is estimated using the relations between Sersic parameters as described by Graham & Driver (2005), with the effective radius $R_e$ and Sersic index $n$ and the ratio between the flux of each component and the total flux being input parameters. The maximum extent of each component is limited to 10 scale-lengths (i.e., $10 \times R_e$). The total number of incident photons per frame is the Poisson variate of $flux_i \times$ the exposure time. The position of each photon in the scene is calculated by sampling the Sersic profiles in polar coordinates. The radial component ($r$) is obtained sampling the integrated profile from which the semi-major and semi-minor axes are derived as

$$\begin{aligned} a &= r/\sqrt{axial\ ratio} \\ b &= a * axial\ ratio \end{aligned} \tag{2}$$

and the azimuthal position ($\chi$) by uniform sampling between [0,2$\pi$). The (x,y) positions of the individual photons on the scene are then

$$\begin{aligned} x &= a * cos(\chi) * cos(\gamma) + b * sin(\chi) * sin(\gamma) \\ y &= -a * cos(\chi) * sin(\gamma) + b * sin(\chi) * cos(\gamma) \end{aligned} \tag{3}$$

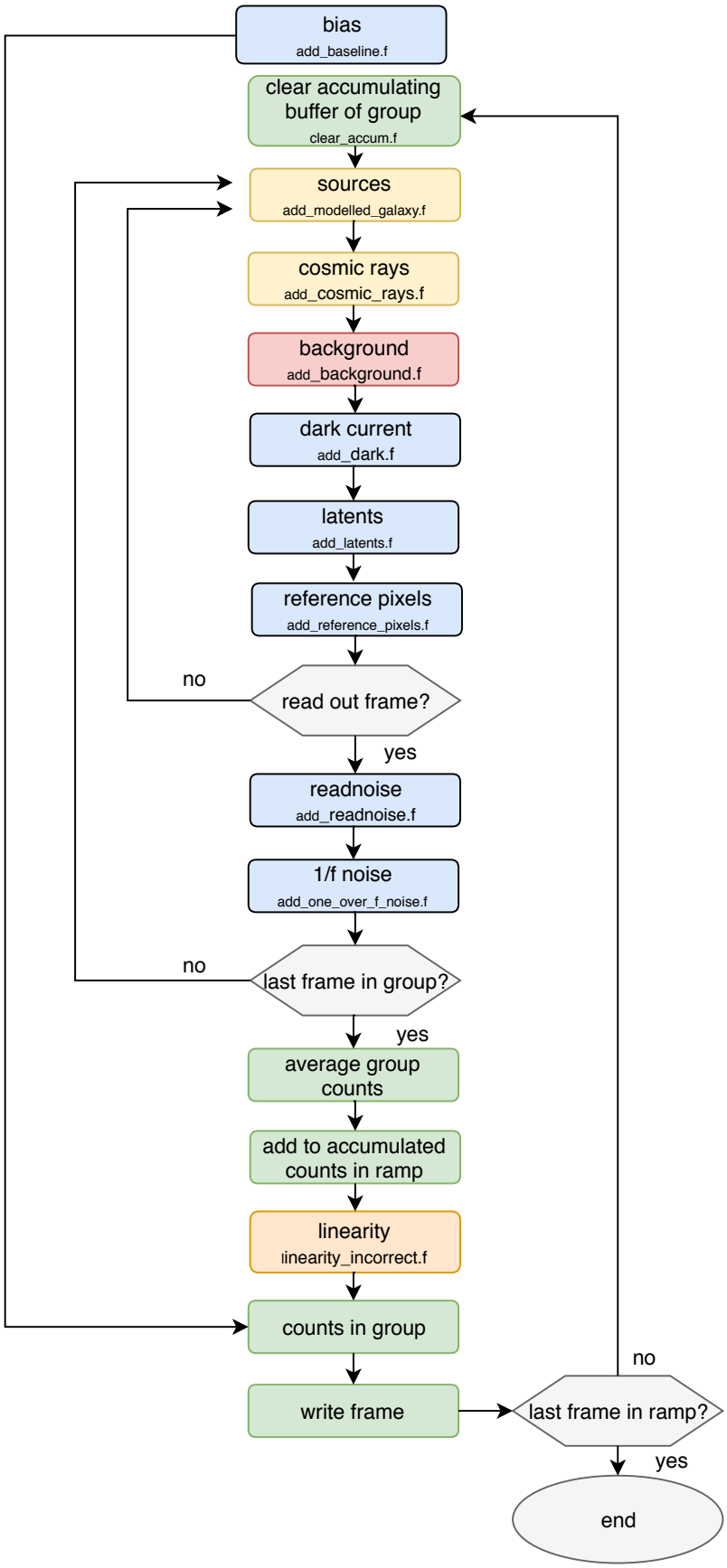


**Figure 2.** Flow chart for *add_up_the_ramp.f*, which is the main routine of *Guitarra*. The blue colors represent detector noise sources, red the noise due to zodiacal background. Green colors represent operations on the images.

where the $\gamma$ = (position angle)$_{jwst}$ + (position angle)$_{galaxy}$. The (x, y) positions are then perturbed by the PSF and the IPC.

Cosmic rays are added by routine *add_modelled_cosmic_rays.f* which first determines the number of events based on the estimated average number of cosmic ray events per frame readout and then samples the cosmic ray matrices calculated by Robberto (2010). The number of events fluctuates between solar maximum (lowest rate) and the solar minimum which is $\sim$ about 3 $\times$ higher than at solar maximum. The user can also chose using the solar flare scenario which has an average level about 1500 $\times$ higher than at solar maximum (Robberto 2010), though in this case cosmic rays completely dominate the scene (and the telescope would be placed in safe mode anyway).

Following the galaxies, stars and cosmic rays, the estimated background due to zodiacal light and thermal background is added on pixel-by-pixel basis, using as input the estimated counts coming from the STScI JWST Backgrounds Tool (https://jwst-docs.stsci.edu/display/JPP/Backgrounds+Tool) for the (estimated) date of observation. The dark current is added at this point using Gaussian variates based on the averages and dispersions measured during ISIMCV3 for individual detectors.

If set, the latent charge is calculated taking into account the decay since the end of the previous exposure and uses the exponential decay rate calculated by Leisenring et al. (2016). The electrons due to latent charge are *not* random-sampled.

The signal of reference pixels is added by *add_reference_pixels.f* that takes into account the different relative responses of pixels, and which also show an eight-pixel periodicity (Table 4). Since the reference pixels typically have an average noise level of 0.8 $\times$ the readnoise of light-sensitive pixels (Rauscher 2015), all regions corresponding to reference pixels have the readnoise $\times$ 0.8 $\times$ responses added using Gaussian variates.

If the frame is not being read out once a cycle is completed, the process returns to adding photons of sources. When the is read out, the readnoise is added by *add_read_noise.f* using Gaussian variates and the readnoise estimated for the individual detectors during ISIMCV3 (Table 6 ) and Gaussian sampling. If 1/f noise is set, all other sources of instrument noise are turned off and the values for reference pixels, readout noise and instrument bias are set by the nghxrg python scripts of Rauscher (2015) as adapted for NIRCam properties by J. Leisenring. All of these are are added on a pixel-by-pixel basis by *add_1_over_f_noise.f*

For the last frame in a group, values in the accumulating buffer are averaged (as will be done in flight), get added to the total count accumulator and are remapped by using the inverse of the linearity correction; at this point the accumulated counts in $e^-$ get converted into ADU using the average gain values measured during ISIMCV3. The baseline image (in **ADU**) derived from the bias image is then added, after which the group is written to the data cube. When this is not the last group in a ramp, the accumulating buffer is cleared and the process of adding charge starts over. If the frame is the last one of the ramp being readout, the output data cube is closed and *Guitarra* completes the execution.

## 4. WRAPPER SCRIPTS

As mentioned above, the files that are input to *Guitarra* are processed by wrapper scripts which come into two languages - *perl* and *python*. The *perl* scripts that do this preparation are *write_jades_parameters.pl* (to read APT output) and *write_parameters.pl* (for custom simulations). The *python* wrappers are *guipytarra.py* and *guitarra_params.py*.

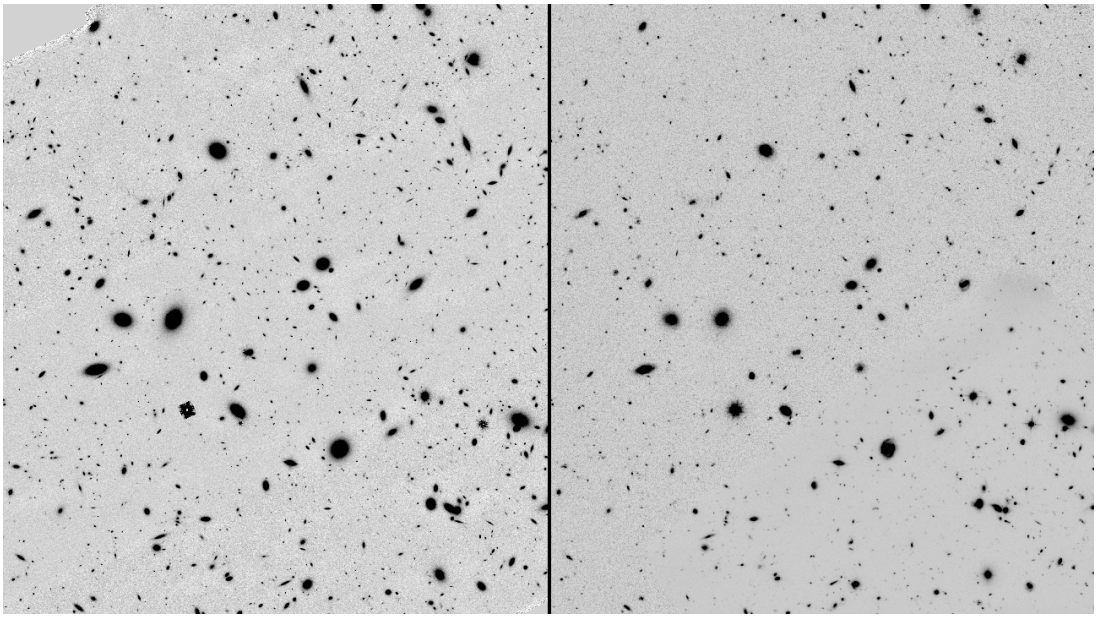

**Figure 3.** Comparison between the output from *Guitarra* , using the JWST F150W filter (left panel) and the HUDF WFC3 mosaic in F160W prepared by the CANDELS team (Grogin et al. 2011; Koekemoer et al. 2011) on the right panel. The simulated image used a catalog of sources with structural parameters calculated by van der Wel et al. (2012). The NIRCam image is for a single exposure using the DEEP8 six group readout mode. All sources in the NIRCam image correspond to catalogued galaxies in van der Wel et al. (2012). The increase in contrast is clearly seen in the NIRCam image, due the higher sensitivity of JWST. On the other hand substructure such as spiral arms are not seen in the NIRCam scene because the simulation used plain Sersic profile parameters.

For either set of scripts, the user must edit them to select which filters will be used in the simulation, and the file with the zodiacal background, which is output by the JWST Backgrounds Tool. These scripts also allow the user to turn on or off the different sources of noise (e.g., dark, cosmic rays, latents). When the scripts are run, they examine the catalog and identify the filters it contains and if these have been flagged to be used, they are selected for inclusion in the output files.

Both scripts create a file with the instrument parameters. The *perl* scripts write a batch file that is used to run the parallel mode script (*parallel.pl*) which will associate to each processor the steps to process a given combination of SCA, filter and dither position.

The batch file contains a sequence of three commands per line, each line being associated to an individual detector and these must run sequentially. The first command selects a sub-catalog of sources centered on each detector. Using the coordinates and the position angle of the NIRCam center, and the SCA being simulated, the catalog coordinates are converted into detector X, Y positions by the *proselytism* program. Objects that fall within 200 pixels of any of the detector borders are included in this catalog, insuring their outer regions are contained in the scene. The second command is the ascii input file to *Guitarra* itself, and is followed by the command to run the *ncdhas* reduction (described below). The parameter-writing script also compiles the source code for *proselytism* and *Guitarra* to insure the executable is current.

## 5. AFTER *Guitarra*

After the data cube is stored, the next step is to calculate the slope image using either the University of Arizona *ncdhas* written by K. Misselt or the reduction code developed at STScI. Both codes remove the instrumental signatures, correct for cosmic ray hits (if set) and make the linearity correction.

A comparison between a scene calculated with *Guitarra* with a real image by the CANDELS team (Grogin et al. 2011; Koekemoer et al. 2011) is shown in Fig. 3. The left panel is a reduced single DEEP8 ramp of 6 groups using the F150W filter. The right panel is a section of the Hubble Ultra Deep Field drizzled mosaic in F160W (Koekemoer et al. 2011). The NIRcam scene is a field reconstruction using the van der Wel et al. (2012) structural catalog and the higher sensitivity of NIRCam and larger mirror area of JWST enhance the contrast between objects and the background relative the the HST data. On the other hand there are some limitations in the NIRCam mock image such as the loss of galaxy substructures due to the use of Sersic profile fits rather than "cloned" images (Bouwens et al. 1998). Another limitation is the non-optimal representation of the JWST/NIRCam PSF due to its very large extent, which is apparent in the two bright stars in the field, both of which are truncated.

When two or more filters and two or more SCAs are used in the simulation the batch file is structured such that there are 3 loops, the innermost one cycling over the dither positions, the middle one over the SCAs and the outermost over the filters. This choice was made to enable covering the surveyed field with at least one detector in a given filter.

## 6. INSTALLATION

*Guitarra* is available at https://github.com/cnaw/guitarra. The *FORTRAN* code is compiled using makefiles for the *Guitarra* main code and *proselytism*. The main code requires the *cfitsio* library which is available from NASA HEARSAC.

To run *Guitarra*, several data files are required - the calibration images used to make the detector footprint; a library of PSF files calculated using WebbPSF (Perrin et al. 2012); the Robberto (2010) cosmic ray library; the set of JWST+NIRCam sensitivity curves; the output from the STScI JWST

**Table 2.** dark current, mean and dispersion, Readout noise levels by **?**

| SCA | $e^-$ | $e^-$ | $e^-$ |
|---|---|---|---|
| A1 | 0.001 | 0.002 | 11.3 |
| A2 | 0.003 | 0.005 | 10.5 |
| A3 | 0.003 | 0.003 | 10.2 |
| A4 | 0.003 | 0.002 | 10.3 |
| A5 | 0.033 | 0.006 | 8.9 |
| B1 | 0.002 | 0.002 | 11.5 |
| B2 | 0.001 | 0.002 | 12.7 |
| B3 | 0.001 | 0.003 | 11.3 |
| B4 | 0.001 | 0.004 | 12.0 |
| B5 | 0.040 | 0.004 | 10.5 |

**Table 3.** Random sampling modes

| Parameter | sampling type |
|---|---|
| background | Poisson |
| dark current | Poisson |
| cosmic rays | Poisson |
| readout noise | Gaussian |
| reference pixels | Gaussian |
| latent charge | none |
| 1/f | none |
| source total counts | Poisson |
| source radial location | uniform |
| source azimuth | uniform |
| PSF | uniform |
| IPC | uniform |

background tool and tables compiled from the SIAF measurements that relate positions on the individual detectors with the V2, V3 positions on the JWST focal plane (NIRCam_flight_transforms.dat), required to make the basic (RA,DEC) to detector (x, y) transformation. An environment variable (GUITARRA_AUX) that points to the directory containing these supporting data needs to be defined so *Guitarra* runs successfully.

There are two contributed (stand-alone) *python* scripts that can be used in the simulations. These are *ng_hxrg_noise* which adds detector noise and is an adaptation of the Rauscher (2015) noise simulator fine tuned to the NIRCam detector properties done by J. Leisenring; the second is the *nrc_distortion_v1.py* script by B. Hilbert that uses the optical model for the JWST focal plane measured from ISIMCV3 to convert positions on the sky to detector (x, y) positions. The perl code requires the $Parallel-ForkManager$ available at cpan to run the *parallel.pl* script, and the standard $Time$ abd $MCE$ modules. The $Astro-FITS-CFITSIO$ *perl* library is required to run the *nchas.pl* wrapper script for the data reduction.

**Table 4.** Reference pixel modulation calculated using a ?? analysis by ??

| Pixel | factor |
|---|---|
| 1 | 1.000000 |
| 2 | 1.000000 |
| 3 | 1.006820 |
| 4 | 1.006820 |
| 5 | 0.983038 |
| 6 | 0.983038 |
| 7 | 1.011640 |
| 8 | 1.011640 |

CNAW thanks D. Erb for providing a digital copy of Q2343-BX1418's spectrum used in the PSF simulations and Dan Kiminki for trying out the first versions of *Guitarra*. Funding from the JWST/NIRCam contract NAS5-02015 to the University of Arizona, the use of the NASA/SAO ADS and the Mikulski Archive for Space Telescopes are gratefully acknowledged.

## APPENDIX

The noise generation with Rauscher's (2015) code includes the detector bias pattern, modeled as white noise, correlated and uncorrelated pink noise, the latter being calculated for each detector amplifier, readnoise and dark current. The white noise is calculated using a Gaussian distribution in a 2048 × 2048 × Nz cube, where Nz is the number of frames (includes frames read and skipped). The pink noise is also derived from (an independent sample of) the white noise, but this time using a 2048 × 2048 × Nz × 2 cube and calculated for individual amplifiers. This corresponds to about 8.6 Gb memory for a DEEP8 pattern with 7 readouts. The pink noise estimates (both correlated and uncorrelated) use direct and inverse FFTs in the noise calculation. These are significantly sped up when using multiple cores (48 to 19 seconds) Bench marking shows that creating the white noise array is another activity with large time impact requiring about 26 seconds to produce a 2048 × 2048 × 128 array.

## REFERENCES

Beers, T. C., Flynn, K., & Gebhardt, K. 1990, AJ, 100, 32

Bertin, E. 2010, Astrophysics Source Code Library, ascl:1010.068

Bertin, E., Mellier, Y., Radovich, M., et al. 2002, Astronomical Data Analysis Software and Systems XI, 281, 228

Bouwens, R., Broadhurst, T., & Silk, J. 1998, ApJ, 506, 557

Erb, D. K., Pettini, M., Shapley, A. E., et al. 2010, ApJ, 719, 1168

Graham, A. W., & Driver, S. P. 2005, PASA, 22, 118

Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, ApJS, 197, 35

Hilbert, B., Canipe, A. M., Robberto, M., & NIRCam Team at STScI 2017, American Astronomical Society Meeting Abstracts #230, 230, 114.01

Kinney, A. L., Calzetti, D., Bohlin, R. C., et al. 1996, ApJ, 467, 38

Koekemoer, A. M., Faber, S. M., Ferguson, H. C., et al. 2011, ApJS, 197, 36

Leisenring, J. M., Rieke, M., Misselt, K., & Robberto, M. 2016, Proc. SPIE, 9915, 99152N

Perrin, M. D., Soummer, R., Elliott, E. M., Lallo, M. D., & Sivaramakrishnan, A. 2012, Proc. SPIE, 8442, 84423D

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes, Cambridge: University Press, 2nd ed., Chapter 7.2

Rauscher, B. J. 2015, PASP, 127, 1144

Robberto, M. 2010, A library of simulated cosmic ray events impacting JWST HgCdTe detectors, JWST-STScI-001928, SM-12

van der Wel, A., Bell, E. F., Häussler, B., et al. 2012, ApJS, 203, 24

Williams, C. C., Curtis-Lake, E., Hainline, K. N., et al. 2018, arXiv:1802.05272