**Process for running MIRISim and JWST pipeline on NEMESIS output**

**Set up**

**Anaconda**

Install and set up anaconda, see:

<https://docs.continuum.io/anaconda/install/>

Install pip package manager:

**conda install pip**

**Mirisim**

Install mirisim, see:

<https://wiki.miricle.org//bin/view/Public/MIRISim_Public>

On completion, there will be a new directory in your $HOME directory called ‘mirisim’

If you are unsure where your $HOME directory is type:

**cd $HOME**

**ls**

Enter the mirisim directory and list the directories inside it:

**cd $HOME/mirisim**

**ls**

Important:

If there is not a directory called ‘CDP’ in mirisim, type:

**export MIRISIM\_ROOT="$HOME/mirisim"**

**export PYSYN\_CDBS="$MIRISIM\_ROOT/cdbs"**

**conda activate mirisim**

**mirisim --get-cdpfiles**

This will take >10 mins to run.

CDP files are reference files for running mirisim. Mirisim can download these as it runs but this often causes issues.

To activate mirisim environment:

**export MIRISIM\_ROOT="$HOME/mirisim"**

**export PYSYN\_CDBS="$MIRISIM\_ROOT/cdbs"**

**export CDP\_DIR="$MIRISIM\_ROOT/CDP"**

**conda activate mirisim**

To deactivate:

**conda deactivate**

**JWST**

Although you could run the JWST pipeline in any environment (or in no environment at all), astroconda contains many of the packages the JWST script requires to run. To set up astroconda see:

<https://astroconda.readthedocs.io/en/latest/>

To activate the environment, type:

**conda activate astroconda**

Install the latest version of the jwst python module (also installs several other packages):

**pip install –upgrade jwst**

This will take 5 mins to run.

Upgrade the packages; gwcs, asdf, specutils, imexam to the latest version:

**pip install –upgrade [package name]**

Install version 1.1.0 of photutils, jwst version 1.3.3 (latest) module is incompatible with newer versions of photutils:

**pip install photutils==1.1.0**

To activate the environment and point to the reference data in $HOME/crds\_cache (downloaded as part of installing jwst module):

**conda activate astroconda**

**export CRDS\_PATH="$HOME/crds\_cache"**

**export CRDS\_SERVER\_URL=”https://jwst-crds.stsci.edu”**

To deactivate:

**conda deactivate**

**Introduction**

**Mirisim**

The script for mirisim is based on the notebook MIRISim\_uranus.ipynb written by Leigh Fletcher to process NEMESIS data of Uranus. This can be found alongside example data at:

<https://uniofleicester-my.sharepoint.com/personal/lnf2_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Flnf2%5Fleicester%5Fac%5Fuk%2FDocuments%2FCode%2FJWST%2FJWSTMIRISim>

The best place to start with mirisim is by practising the examples in the Appendix section of:

<https://wiki.miricle.org/pub/Public/MIRISimPublicRelease2dot3/MIRISim.pdf>

This document is also a useful source of information should any problems occur.

**JWST pipeline**

The script for running the JWST pipeline stages is based on notes by David Law for running the pipeline on a point source. This can be found on the JWST webinar pages at:

<https://stsci.app.box.com/s/n5pv4pf1gj8ducdrjpd4lf8wiw1mfc7a>

And the notebook can be found at:

<https://github.com/STScI-MIRI/MRS-ExampleNB/tree/main/Notebook2>

**Running MIRISim**

Create a directory (working directory) to run the process in:

**mkdir [name]**

Download:

**runmsim\_v6.py**

and place into above directory

Place the input to mirisim (either NEMESISCubes directory or .fits cube) in the directory

Start the mirisim environment:

**export MIRISIM\_ROOT=”$HOME/[path to mirisim folder from $HOME]”**

**export PYSYN\_CDBS="$MIRISIM\_ROOT/cdbs"**

**export CDP\_DIR="$MIRISIM\_ROOT/CDP"**

**conda activate mirisim**

If you are unsure where $HOME is, type:

**cd $HOME**

**pwd**

Open the mirisim script using nano:

**nano runmsim\_v6.py**

Between lines 2 and 26, observational parameters relating to exposure time, dithering and channels used can be changed to be the same as the values that will be used for the actual JWST observations.

On line 14 it is possible to tell the script whether to run multiprocessing or not by setting the parameter ‘usage’ to ‘None’ for no multiprocessing or; ‘quarter’, ‘half’ or ‘full’ to run multiprocessing using different numbers of CPUs. Note using a large number of CPUs will slow down other processes on your computer and may cause it to crash if too many processes are run at once.

To turn on dithering, set **dither** on line 13 to equal **True**

To change the number of dither positions, set **number\_dithers** on line 14 to the number of dither positions. Options are 2 or 4 (for 2 or 4 point dither pattern)

To change what channel the dither pattern is optimised for, alter the value of **start\_index** on line 15

Point source dither patterns

ALL/Channel 1: start\_index = 1

Channel 2: 9

Channel 3: 17

Channel 4: 25

Extended source dither patterns

ALL: 33

Channel 1: 37

Channel 2: 41

Channel 3: 45

Channel 4: 49

To save the det\_images into a directory named **stage0** (default input to JWST pipeline), set **save0** on line 26 equal to **True**. Otherwise leave as **False**

Close the script:

**{control} o**

**{enter}**

**{control} x**

Run the mirisim script

**python -W ignore runmsim\_v6.py [file]**

Where [file] is the name of your input file, the script will detect if it is a directory of NEMESIS files or a .fits cube. If it is the first, the script will generate a .fits cube from the NEMESIS files, then it will run mirisim on that cube. If it is the latter, the script will just run mirisim on the cube.

If errors/warnings of type: ‘No valid cache found’ consult notes on cache issues

Outputs:

**SHORT\_data**

**MEDIUM\_data**

**LONG\_data**

**stage0** (if **save0 = True**)

In each disperser directory, there will be a directory named: ‘det\_images’. In this will be the simulated MIRI raw data for that particular disperser.

All the det\_image raw images from each disperser will also have been copied to the directory ‘stage0’ If ‘y’ is selected. For 1 exposure:

With no dithering: 6 files should be in stage0

With 2 point dithering: 12 files should be in stage0

With 4 point dithering: 24 files should be in stage0

The script outputs the number of files in stage0, check this is correct.

Deactivate the mirisim environment:

**conda deactivate**

**Running JWST pipeline using previous outputs in stage 0 directory**

Download:

**jwstpipeline\_v3.py**

and place into working directory

Activate astroconda environment and declare directory routes:

**conda activate astroconda**

**export CRDS\_PATH="$HOME/[path to crds\_cache directory from $HOME]"**

**export CRDS\_SERVER\_URL="https://jwst-crds.stsci.edu"**

If you wish to alter multiprocessing from ‘all’:

Open the JWST script using nano:

**nano jwstpipeline\_v3.py**

At the top of the script; usage can be changed (see instructions for changing this in mirisim script)

Run the JWST script:

**python -W ignore jwstpipeline\_v3.py [stage 0 data directory]**

Stage 0 data is raw data from JWST. If you just ran mirisim (above) then stage 0 data directory is ‘stage0’.

Outputs:

**stage1**

**stage2**

**stage3**

For details of what the pipeline does at each stage see:

<https://jwst-pipeline.readthedocs.io/en/latest/jwst/pipeline/index.html>

For stage 1, the script looks for files in stage0 that start with the string: ‘det’ and end with ‘exp1.fits’. This is compatible with the output from mirisim. If your files are different you may need to edit line 235 of the JWST script or change the names of the files in stage0.

The cubes are located in stage3, a 1D spectrum fits cube is generated for each cube. In the case of extended sources, this is produced by averaging the pixel values for each 2D image across the 3D cube with no background subtraction being applied. To plot this file as a spectra, see eg section 8 of:

<https://stsci.app.box.com/s/n5pv4pf1gj8ducdrjpd4lf8wiw1mfc7a>

**Notes**

runmsim\_v6.py creates the directory stage0. If this directory already exists the script will not run to ensure data from 2 different mirisimulations don’t end up in the same directory. Change the name of the directory, move it or delete it.

The JWST pipeline script is set up for extended sources. Significant alteration of the script would be required to convert it to run on point sources.