



Using the IDIA MeerKAT Pipeline

Dr Jordan Collier

Ilifu Support Astronomer, IDIA, Department of Astronomy, University of Cape Town
Adjunct Fellow, Western Sydney University

Bradley Frank, Srikrishna Sekhar, Russ Taylor



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD



Inter-University Institute
for Data Intensive Astronomy

WESTERN SYDNEY
UNIVERSITY





Inter-University Institute
for Data Intensive Astronomy

IDIA & Ilifu



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD



WESTERN SYDNEY
UNIVERSITY



Jordan Collier | 20 Feb 2019 | MIGHTEE workshop

Our Systems: Software Stack

Astronomy Tools & Libraries

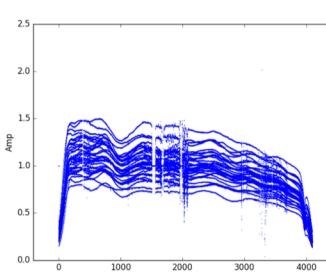
Research Computing

Scientific Libraries

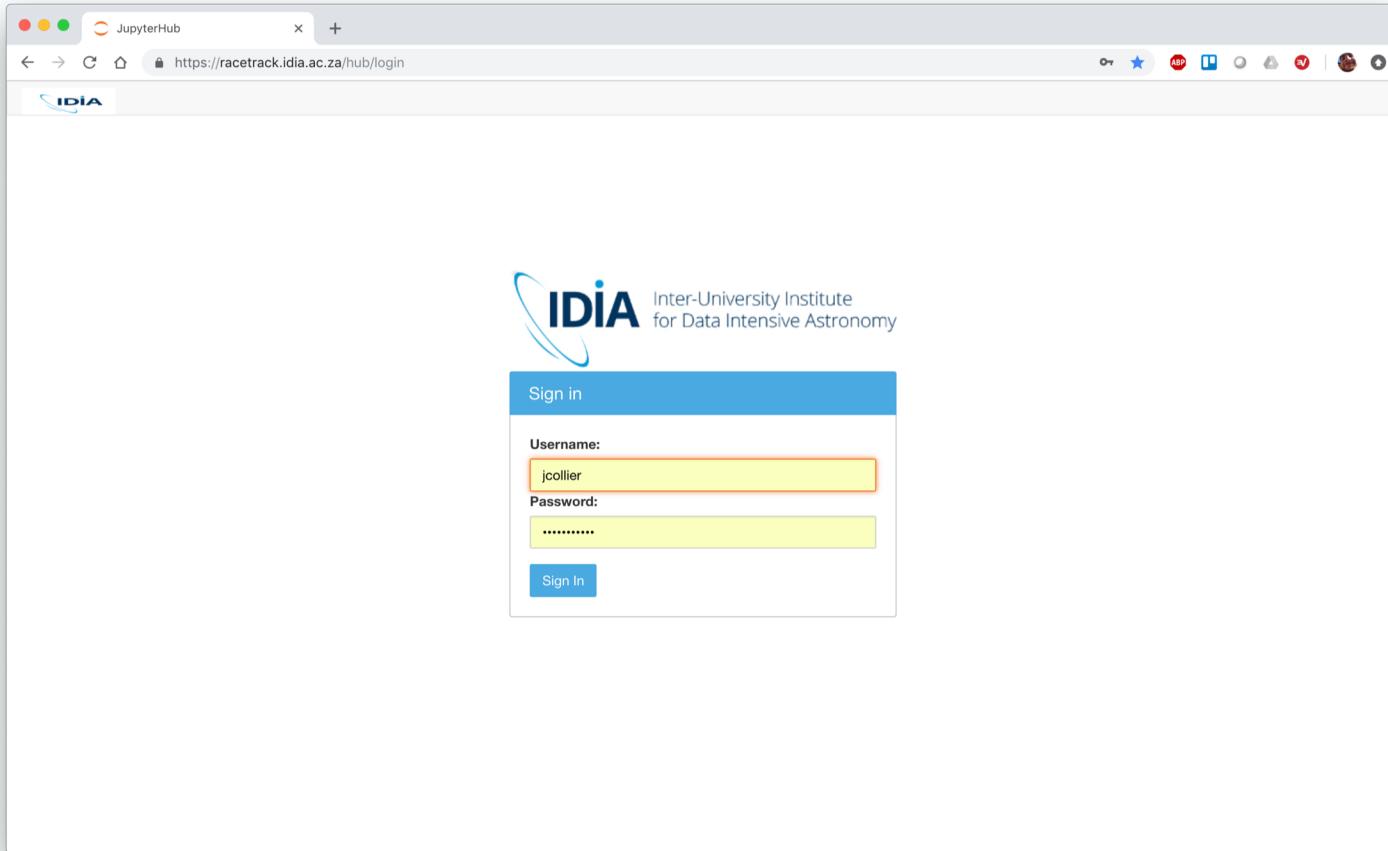
General Purpose Computing



ssh access

```
Warning: No xauth data; using fake authentication data for X11 forwarding.  
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-137-generic x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
Get cloud support with Ubuntu Advantage Cloud Guest:  
http://www.ubuntu.com/business/services/cloud  
  
58 packages can be updated.  
3 updates are security updates.  
  
New release '18.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Mon Nov 26 02:19:05 2018 from 137.154.142.200  
jcollier@racetrack:~$ ls data/CASA_testing/plots/  
jcollier@racetrack:~$ ls data/CASA_testing/plot  
plotcal.last plots.last plots/  
jcollier@racetrack:~$ ls data/CASA_testing/*plots/  
1491550051_bppass_gain.png 1491550051_bppass_real_imag.png 1491550051_fastplot_phase.png 1491550051_gain_phase.png 1491550051_target_spectrum.png  
1491550051_bppass_phase.png 1491550051_fastplot_amp.png 1491550051_gain_amp.png 1491550051_target_spec.pdf  
1491550051_bppass.png 1491550051_fastplot_phase_benchmarking_helo.png 1491550051_gain_Amp.png 1491550051_target_spec.png  
jcollier@racetrack:~$ ls data/CASA_testing/*plots/1491550051_bppass_gain.png  
data/CASA_testing/1491550051/plots/1491550051_bppass_gain.png  
jcollier@racetrack:~$ imgcat data/CASA_testing/*plots/1491550051_bppass_gain.png  
imgcat: command not found  
jcollier@racetrack:~$ imgcat data/CASA_testing/*plots/1491550051_bppass_gain.png  
  
  
jcollier@racetrack:~$
```

JupyterHub



Jordan Collier | 20 Feb 2019 | MIGHTEE workshop



JupyterHub

The screenshot shows a JupyterLab interface with the following components:

- File Browser:** On the left, a sidebar titled "notebooks" lists files and folders. The "ms-vis.ipynb" file is selected.
- Code Editor:** The main area contains a code cell with Python code related to astronomical data processing using CASA. The code includes calls to `msvis` and `ms` functions.
- Terminal Output:** Below the code cell, the terminal output shows the execution of the code, including messages about reading tables and extracting data, along with generated plots.
- Plot:** A scatter plot titled "fastplot.png" is displayed, showing a distribution of data points in a 2D space with axes labeled "Real" and "Imag".

Singularity Containers

- Led by Jeremy Smith (IDIA, UWC) – slide courtesy
- Containers are a software package that contains everything required to run an application/workflow
 - files, environmental variables, libraries and dependencies
- Allows for interchangeable environments and configurations at runtime and portability between systems



Singularity Containers

```
sourcefinding_py3_update.sh — ~/Research/MeerKAT/containers
1 Bootstrap: localimage
2 From: /mnt/containers/sourcefinding_py3_update/sourcefinding_py3-2018-08-30-tmp.simg
3 Include: software-properties-common
4
5 %environment
6
7     export INSTALLER_PATH=/installer
8     ...
9
10 %setup
11     #Copy patch file for blobcat
12     cp blobcat_patch.diff $SINGULARITY_ROOTFS
13
14 %post
15
16     #Installation of BLOBCAT
17     export BLOBCAT_PATH=$INSTALLER_PATH/BLOBCAT
18     mkdir $BLOBCAT_PATH
19     cd $BLOBCAT_PATH
20     wget -O blobcat-v1.2.tar.bz2 http://sourceforge.net/projects/blobcat/files/blobcat-v1.2.tar.bz2/download
21     tar xvjf blobcat-v1.2.tar.bz2
22
23     #Patch blobcat to run python2.7 and fix error on one line
24     patch blobcat.py /blobcat_patch.diff
25
26     ...
27
28     #Upgrade pip
29     pip install --upgrade pip
30
31     #Installation of PASTA
32     pip install git+https://github.com/bwkeller/PASTA
33
34     ...
35
36     #Install eog (and dependencies)
37     apt-get -y install eog
sourcefinding_py3_update.sh  1:1
```

1 LF UTF-8 Shell Script 0 files 4 updates

Singularity Containers

```
1. IPython: users/jcollier (ssh)
>jcollier@racetrack:~$ singularity shell /data/exp_soft/containers/casa-stable-5.3.0.simg
Singularity: Invoking an interactive shell within container...

Singularity casa-stable-5.3.0.simg:~> casa --nologger --nogui --nologfile

=====
The start-up time of CASA may vary
depending on whether the shared libraries
are cached or not.
=====

IPython 5.1.0 -- An enhanced Interactive Python.

CASA 5.3.0-143 -- Common Astronomy Software Applications

--> CrashReporter initialized.
Enter doc('start') for help getting started with CASA...
Using matplotlib backend: TkAgg

CASA <1>: █
```

Singularity Containers

- Exec

```
:~$ singularity exec /data/exp_soft/containers/sourcefinding_py3.simg python myscript.py
```

- MPICASA with SLURM using batch file (“my-casa-job.sh”)

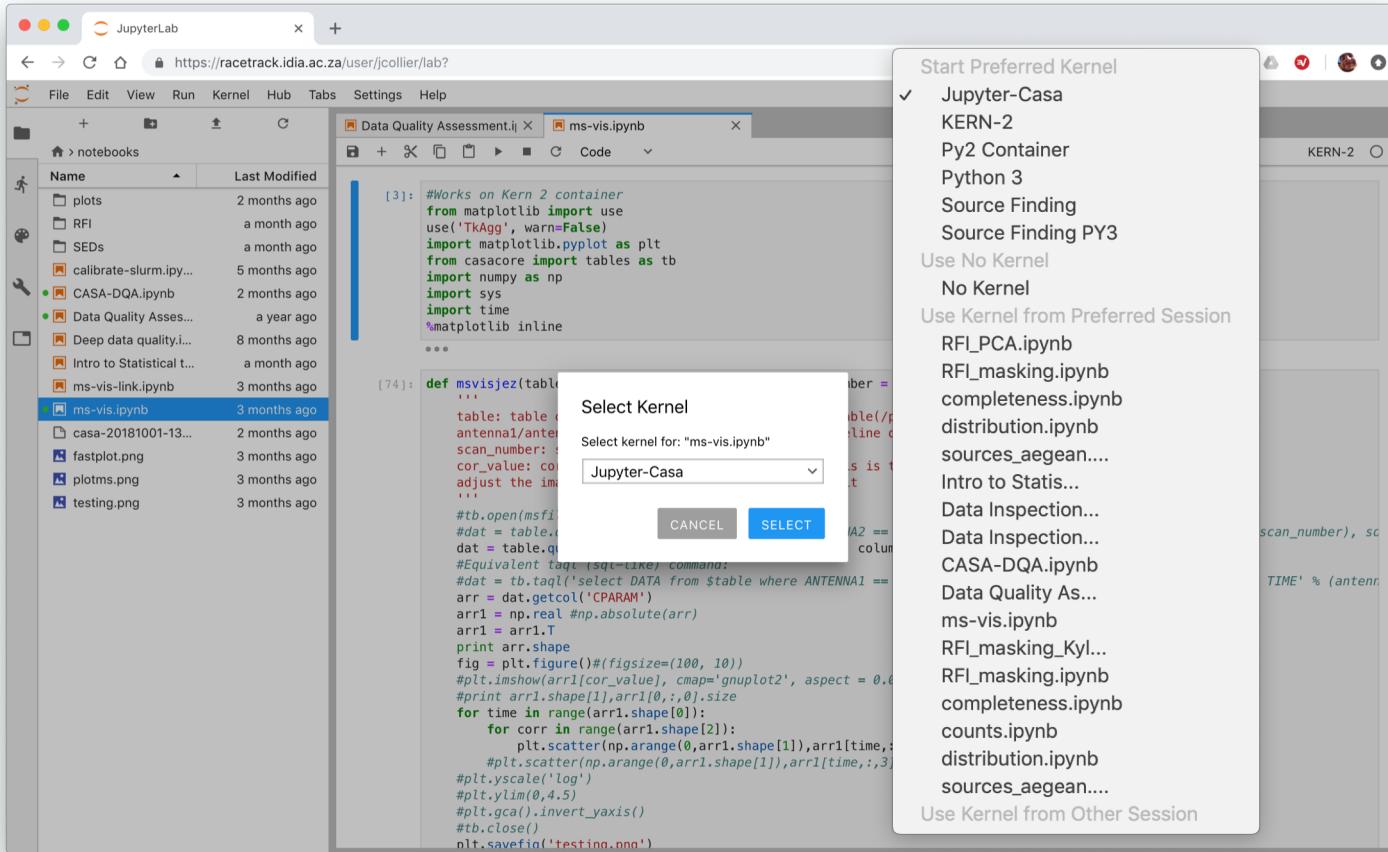
```
#!/bin/bash
#SBATCH --nodes=15
#SBATCH --ntasks-per-node=8
#SBATCH --cpus-per-task=3
#SBATCH --mem=98304
#SBATCH --job-name=tclean
#SBATCH --distribution=plane=4
#SBATCH --output=logs/tclean-%j.out
#SBATCH --error=logs/tclean-%j.err

/path/to/mpicasa /usr/bin/singularity exec /data/exp_soft/pipelines/casameer-5.4.0.simg "casa"
--nologger --nogui --logfile logs/tclean-${SLURM_JOB_ID}.casa -c tclean.py --config .config.tmp
```

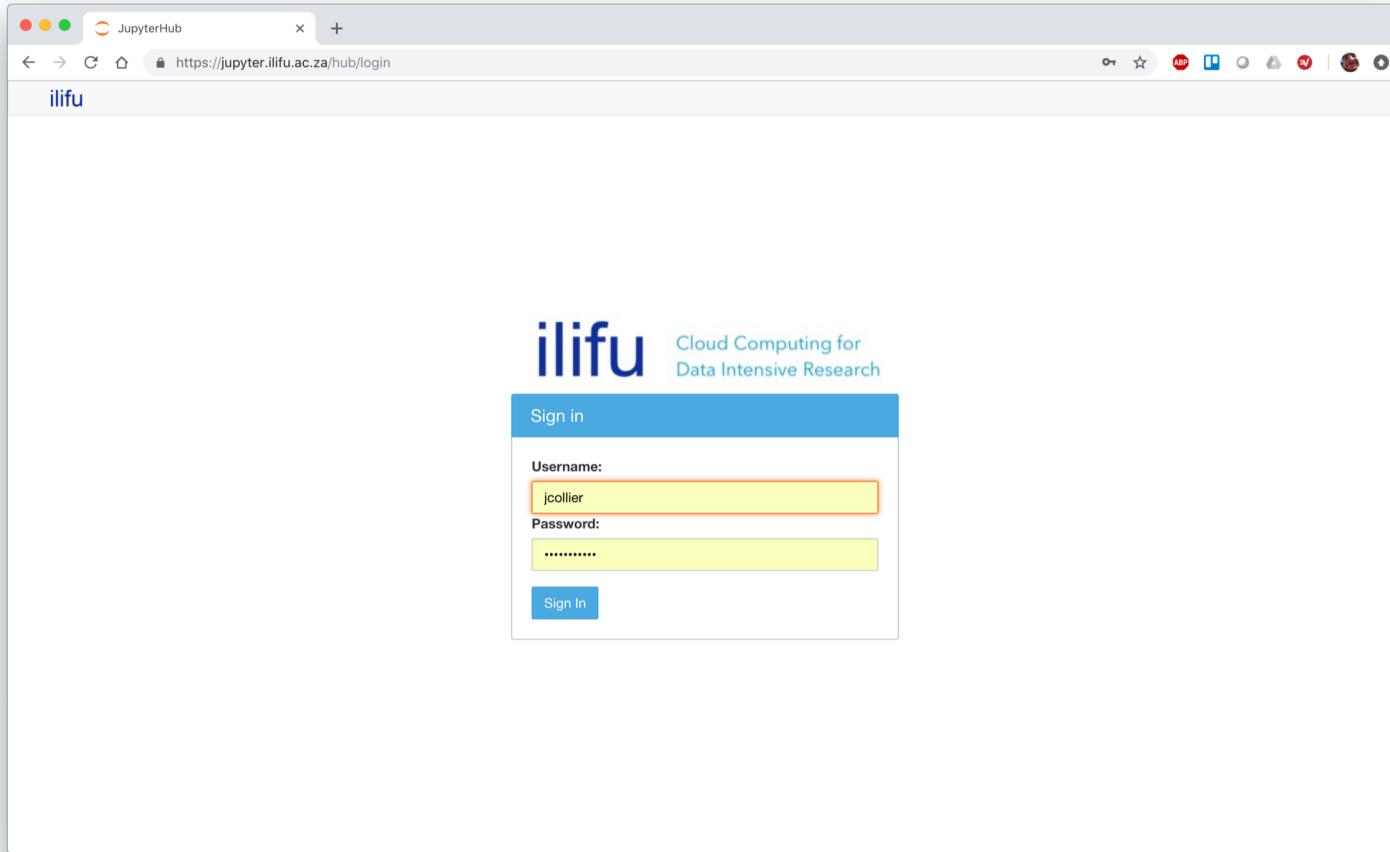
- Terminal on SLURM cluster:

```
user@master001:~> sbatch my-casa-job.sh
```

JupyterHub: Singularity Containers



JupyterHub Spawner



Jordan Collier | 20 Feb 2019 | MIGHTEE workshop



JupyterHub Spawner

Spawner Options

Select a job profile:

Minimum Node - 1 core, 4 GB, 72 hours

Spawn

- ✓ Minimum Node - 1 core, 4 GB, 72 hours
- Small Node - 2 core, 16 GB, 72 hours
- Medium Node - 4 core, 32 GB, 72 hours
- Large Node - 8 core, 64 GB, 72 hours
- Half-Max Node - 16 core, 125 GB, 72 hours
- Max Node - 32 core, 250 GB, 72 hours
- Max High Memory Node - 32 core, 500 GB, 72 hours



Inter-University Institute
for Data Intensive Astronomy

Using the IDIA Pipeline



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

WESTERN SYDNEY
UNIVERSITY



Jordan Collier | 20 Feb 2019 | MIGHTEE workshop

processMeerKAT.py

- Builds and submits pipeline job on IDIA / Ilifu SLURM cluster
- Launch pipeline
 - Input measurement set
 - Build your config file
 - OR run your config file (i.e. build job scripts)
 - Request resources
 - Optionally insert your own scripts
 - Specify containers and MPI wrappers

```
MC02RTBRAG&WP:~ jordan$ processMeerKAT.py -h
usage: /Users/jordan/Research/MeerKAT/IDIA/pipelines/processMeerKAT/processMeerKAT.py
      [-h] [-M path] [-C path] [-N num] [-t num] [-P num] [-m num] [-p name]
      [-T time] [-S script threadsafe container] [-w path] [-c path] [-l]
      [-s] [-v] (-B | -R | -V)

Process MeerKAT data via CASA measurement set. Version: 1.0

optional arguments:
  -h, --help            show this help message and exit
  -M path, --MS path    Path to measurement set.
  -C path, --config path
                        Path to config file.
  -N num, --nodes num   Use this number of nodes [default: 8; max: 35].
  -t num, --ntasks-per-node num
                        Use this number of tasks (per node) [default: 4; max:
                        128].
  -P num, --plane num   Distribute tasks of this block size before moving onto
                        next node [default: 2; max: ntasks-per-node].
  -m num, --mem num     Use this many GB of memory (per node) for threadsafe
                        scripts [default: 236; max: 236].
  -p name, --partition name
                        SLURM partition to use [default: 'Main'].
  -T time, --time time  SLURM partition to use [default: 'Main'].
  -S script threadsafe container, --scripts script threadsafe container
                        Run pipeline with these scripts, in this order, using
                        this container (3rd value - empty string to default to
                        [-c --container]). Is it threadsafe (2nd value)?
  -w path, --mpi_wrapper path
                        Use this mpi wrapper when calling threadsafe scripts
                        [default: '/data/exp_soft/pipelines/casa-
                        prerelease-5.3.0-115.el7/bin/mpicasa'].
  -c path, --container path
                        Use this container when calling scripts [default:
                        '/data/exp_soft/pipelines/casameer-5.4.1.xvfb.simg'].
  -l, --local           Build config file locally (i.e. without calling srun)
                        [default: False].
  -s, --submit          Submit jobs immediately to SLURM queue [default:
                        False].
  -v, --verbose         Verbose output? [default: False].
  -B, --build           Build config file using input MS.
  -R, --run              Run pipeline with input config file.
  -V, --version         Display the version of this pipeline and quit.
```

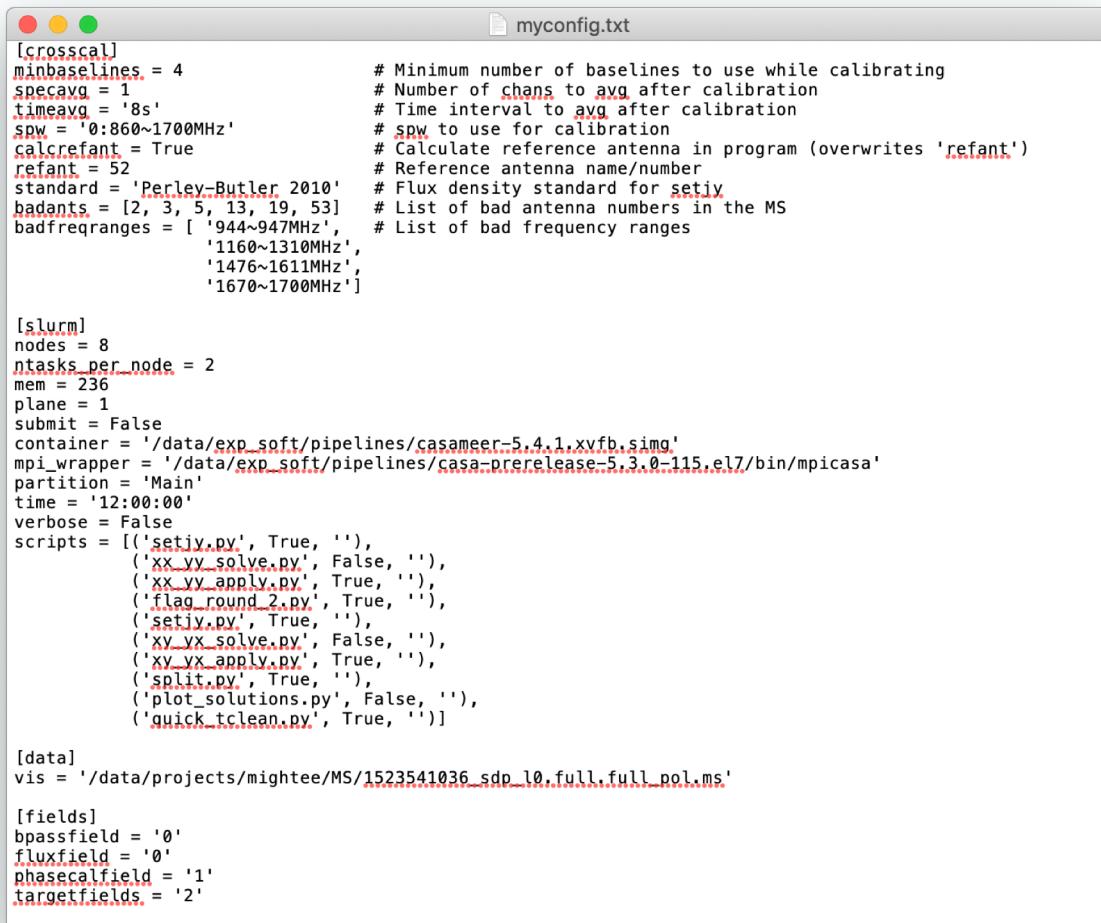
The Build Step

- `processMeerKAT.py -B -C myconfig.txt -M /data/projects/mightee/MS/1524147354_sdp_10.full.full_pol.MS`
- Builds your config file
 - Reads MS and automatically extracts field IDs based on intent
 - Extracts total flux, bandpass, phase cal, and targets
 - Compares resources you've requested to the number of scans / sub-MSs
 - We partition by scan, so it prevents requesting too many / few threads
 - Ensures reference antenna exists



Config Files

- [crosscal]
 - Select spectral window / freqs (e.g. for HI)
 - Other parameters affecting calibration
- [slurm]
 - Resource parameters
 - List of scripts to run, and whether to call MPI
 - etc
- Path to MS stored
- Field IDs stored



```
[crosscal]
minbaselines = 4
specavg = 1
timeavg = '8s'
spw = '0:860~1700MHz'
calrefant = True
refant = 52
standard = 'Perley-Butler 2010'
badants = [2, 3, 5, 13, 19, 53]
badfreqranges = [
    '944~947MHz',
    '1160~1310MHz',
    '1476~1611MHz',
    '1670~1700MHz']

[slurm]
nodes = 8
ntasks_per_node = 2
mem = 236
plane = 1
submit = False
container = '/data/exp_soft/pipelines/casameer-5.4.1.xvfb.simg'
mpi_wrapper = '/data/exp_soft/pipelines/casa-prerelease-5.3.0-115.el7/bin/mpicasa'
partition = 'Main'
time = '12:00:00'
verbose = False
scripts = [
    ('setijy.py', True, ''),
    ('xx_yy_solve.py', False, ''),
    ('xx_yy_apply.py', True, ''),
    ('flag_round_2.py', True, ''),
    ('setijy.py', True, ''),
    ('xv_xy_solve.py', False, ''),
    ('xy_xv_apply.py', True, ''),
    ('split.py', True, ''),
    ('plot_solutions.py', False, ''),
    ('quick_tclean.py', True, '')]

[data]
vis = '/data/projects/mightee/MS/1523541036_sdp_l0.full.full.pol.ms'

[fields]
bpassfield = '0'
fluxfield = '0'
phasescalfield = '1'
targetfields = '2'
```

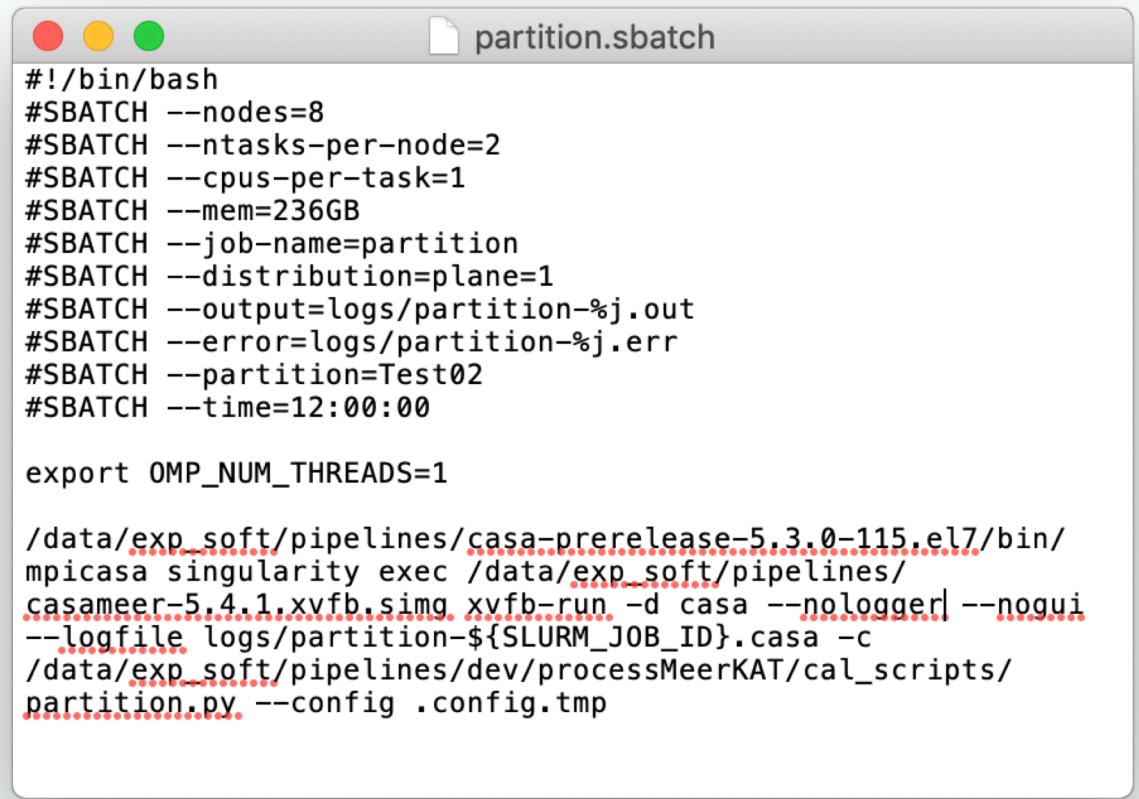
The Run Step

- `processMeerKAT.py -R -C myconfig.txt`
- Builds SLURM sbatch files
(one for each unique python script)
 - Wrote sbatch file "validate_input.sbatch"
 - Wrote sbatch file "partition.sbatch"
 - Wrote sbatch file "calc_refant.sbatch"
 - Wrote sbatch file "flag_round_1.sbatch"
 - Wrote sbatch file "setjy.sbatch"
 - Wrote sbatch file "xx_yy_solve.sbatch"
 - Wrote sbatch file "xx_yy_apply.sbatch"
 - ...
 - Wrote sbatch file "split.sbatch"
 - Wrote sbatch file "plot_solutions.sbatch"
 - Wrote sbatch file "quick_tclean.sbatch"
- Writes master submission (bash) script



Sbatch Files

- Calls script within CASA container, within MPI wrapper
- Logs with job ID written for stdout, stderr, and CASA
- User can choose to submit these jobs one at a time
- OR use master submission script



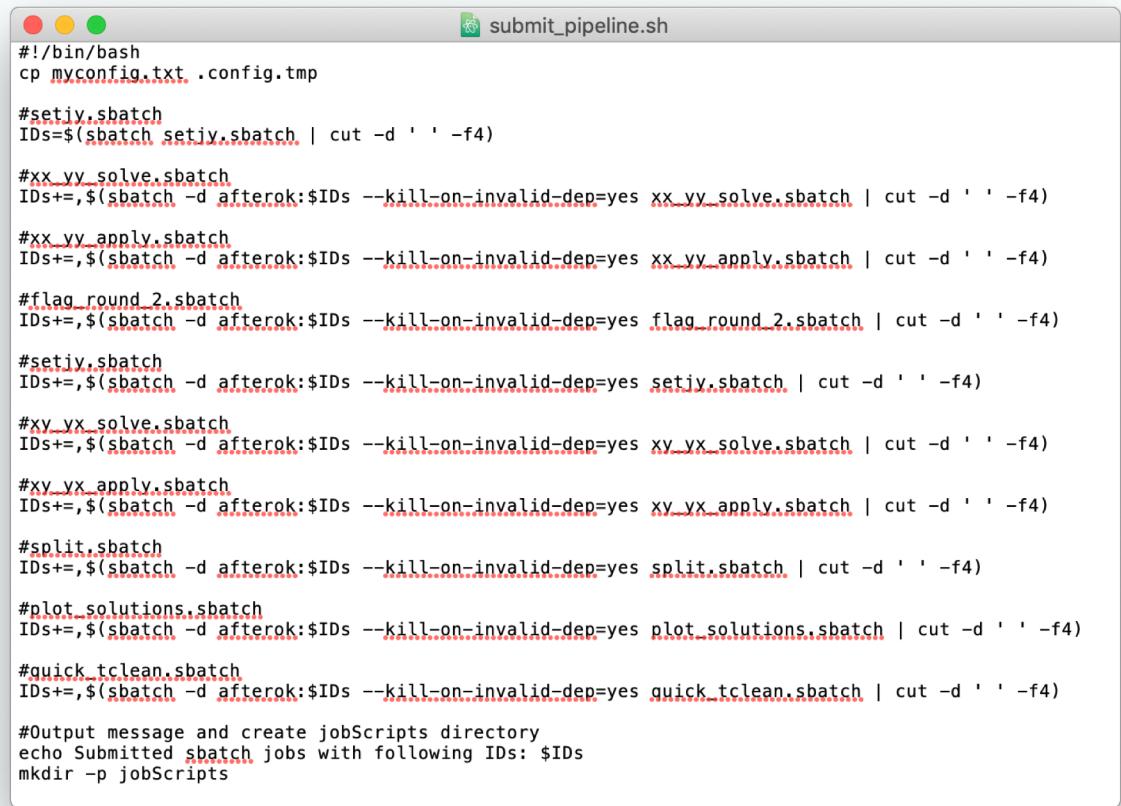
```
#!/bin/bash
#SBATCH --nodes=8
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=1
#SBATCH --mem=236GB
#SBATCH --job-name=partition
#SBATCH --distribution=plane=1
#SBATCH --output=logs/partition-%j.out
#SBATCH --error=logs/partition-%j.err
#SBATCH --partition=Test02
#SBATCH --time=12:00:00

export OMP_NUM_THREADS=1

/data/exp_soft/pipelines/casa-prerelease-5.3.0-115.el7/bin/
mpicasa singularity exec /data/exp_soft/pipelines/
casameer-5.4.1.xvfb.simg xvfb-run -d casa --nologger --nogui
--logfile logs/partition-${SLURM_JOB_ID}.casa -c
/data/exp_soft/pipelines/dev/processMeerKAT/cal_scripts/
partition.py --config .config.tmp
```

Master Submission Script

- Bash script that submits sbatch files to SLURM queue each time it's run
- Builds ancillary bash scripts that interact with pipeline run
 - Summarise progress
 - Kill all jobs
 - Find errors (after pipeline run)
 - Display runtimes (after pipeline run)



```
#!/bin/bash
cp myconfig.txt .config.tmp

#setiy.sbatch
IDs=$(sbatch setiy.sbatch | cut -d ' ' -f4)

#xx_yy_solve.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes xx_yy_solve.sbatch | cut -d ' ' -f4)

#xx_yy_apply.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes xx_yy_apply.sbatch | cut -d ' ' -f4)

#flag_round_2.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes flag_round_2.sbatch | cut -d ' ' -f4)

#setiy.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes setiy.sbatch | cut -d ' ' -f4)

#xy_yx_solve.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes xy_yx_solve.sbatch | cut -d ' ' -f4)

#xy_yx_apply.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes xy_yx_apply.sbatch | cut -d ' ' -f4)

#split.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes split.sbatch | cut -d ' ' -f4)

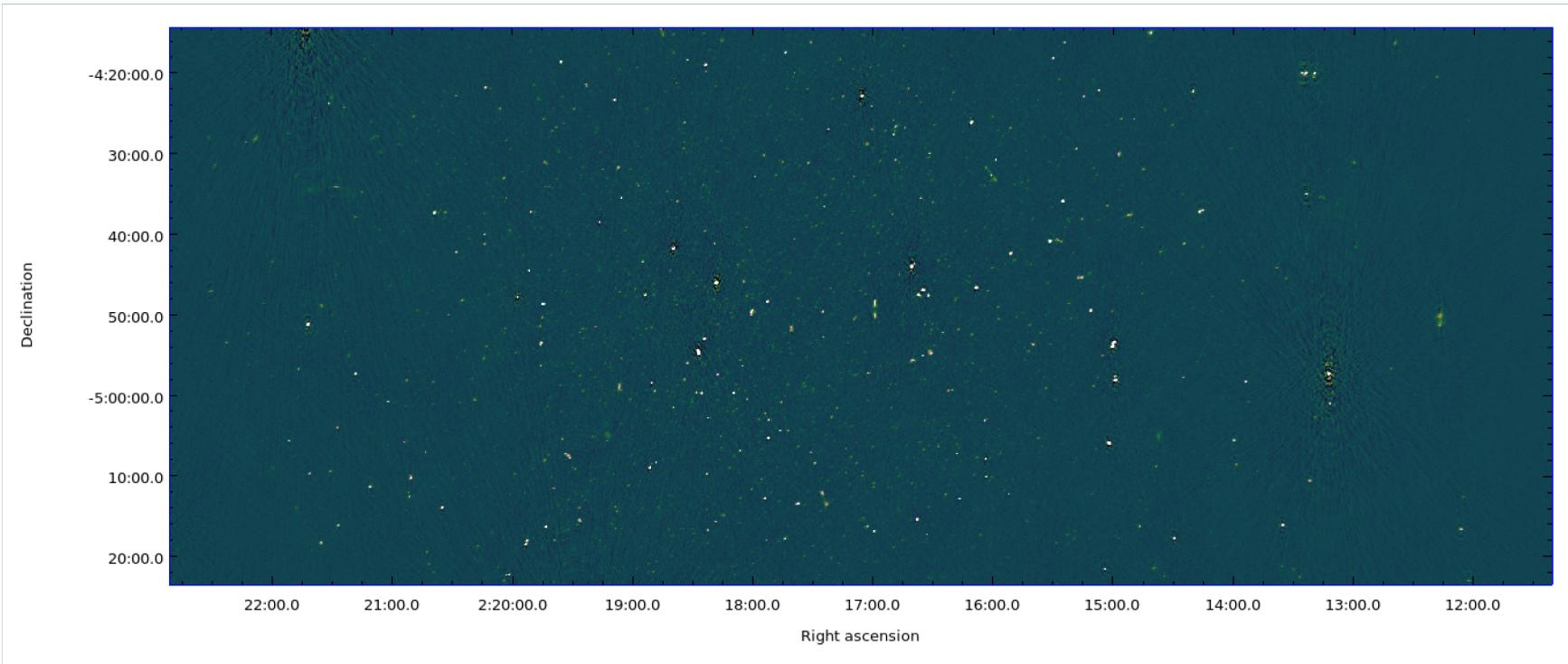
#plot_solutions.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes plot_solutions.sbatch | cut -d ' ' -f4)

#quick_tclean.sbatch
IDs+=,$(sbatch -d afterok:$IDs --kill-on-invalid-dep=yes quick_tclean.sbatch | cut -d ' ' -f4)

#Output message and create jobScripts directory
echo Submitted sbatch jobs with following IDs: $IDs
mkdir -p jobScripts
```

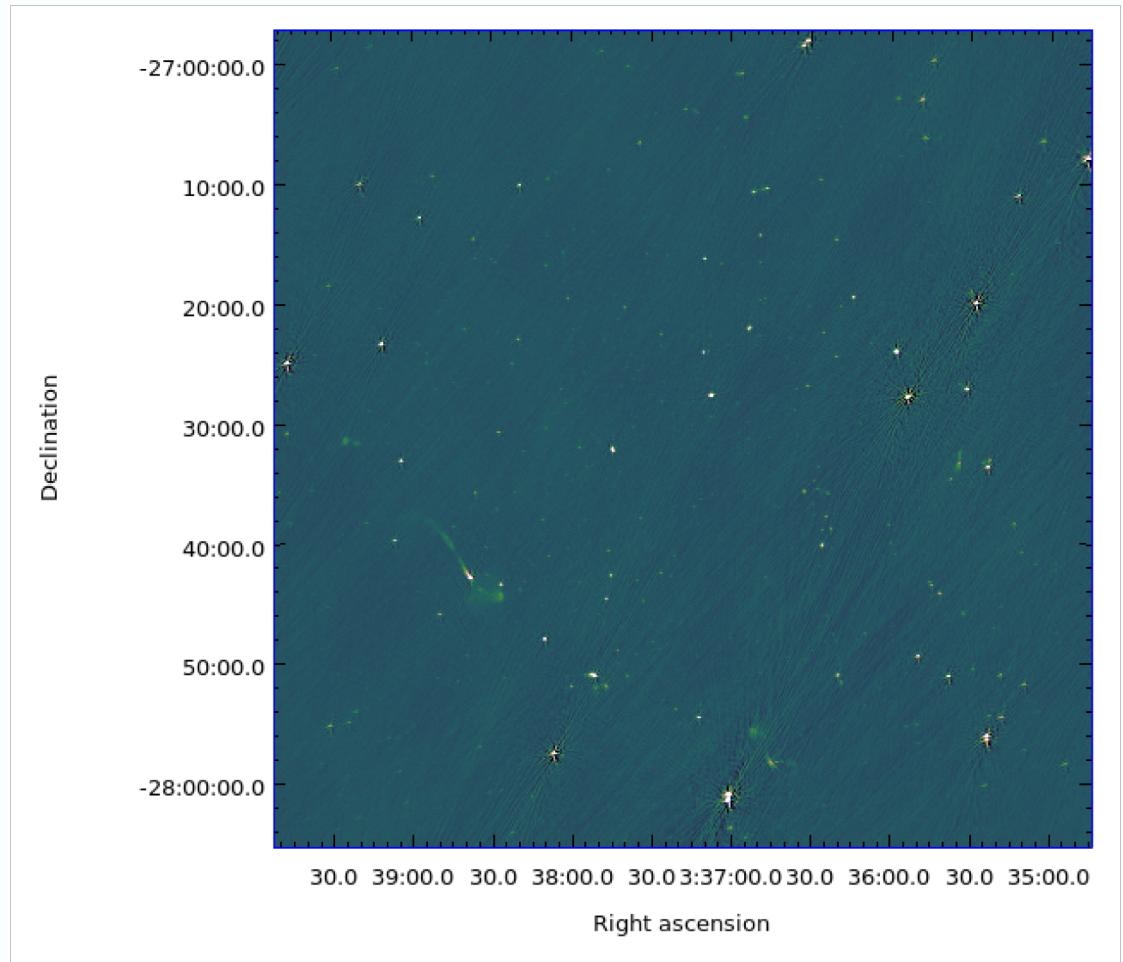
Initial QA: Quick Look Images

- Very quick and dirty imaging for QA purposes
 - No selfcal, no w-projection, no thresholding, no multi-scale, etc
 - XMM-LSS field: RMS ~10 uJy / beam



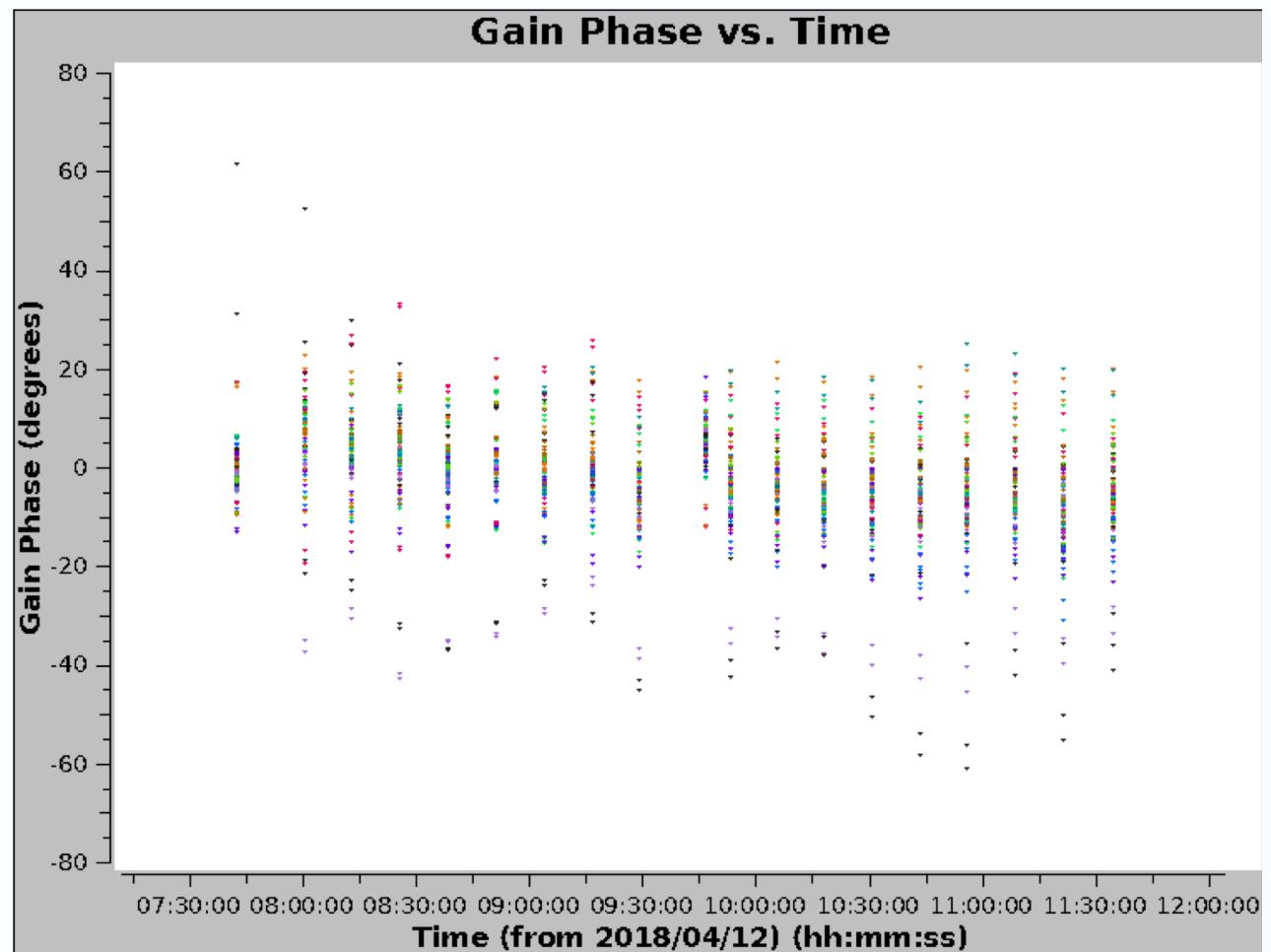
Initial QA: Quick Look Images

- CDFS field
- ~4 hours,
50 MHz spw
- RMS ~80 uJy / beam
- Scales to ~10 uJy over whole band (in 4 hours)



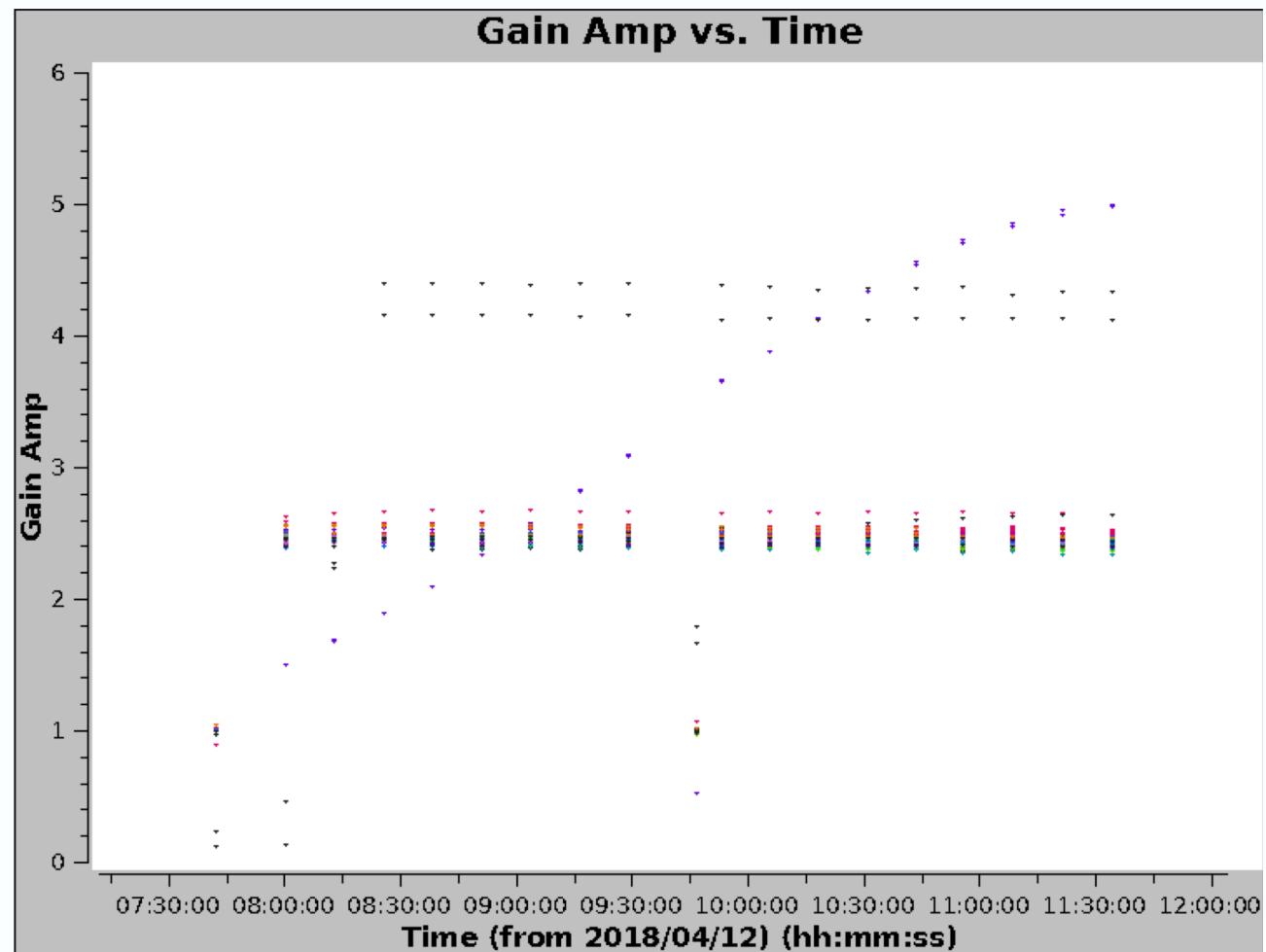
Initial QA: Calibration Solutions

- Phase cal solutions



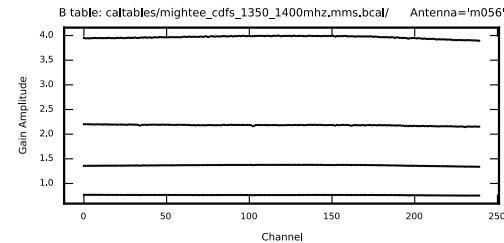
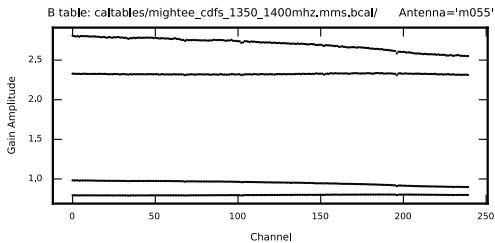
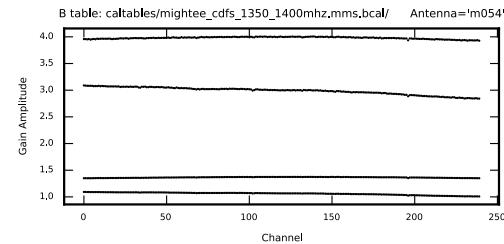
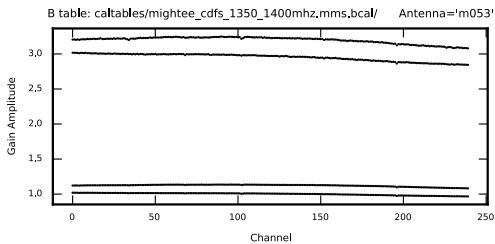
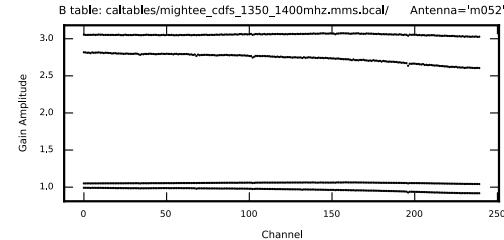
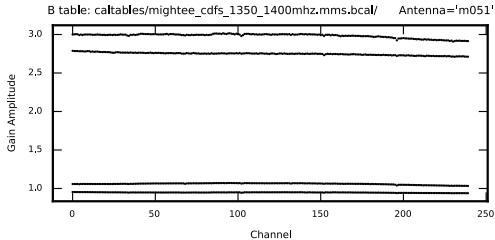
Initial QA: Calibration Solutions

- Phase cal solutions



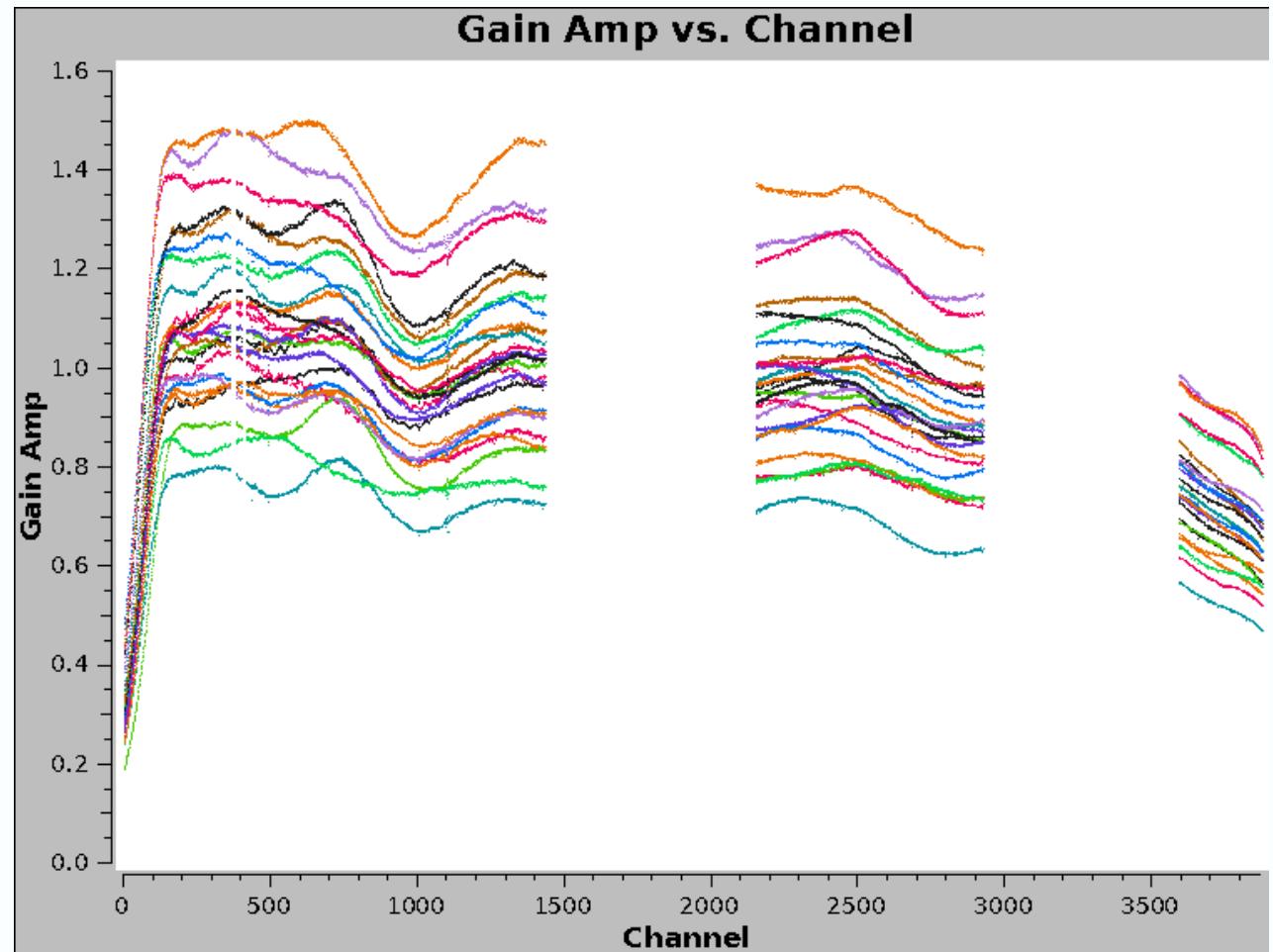
Initial QA: Calibration Solutions

- Phase cal solutions



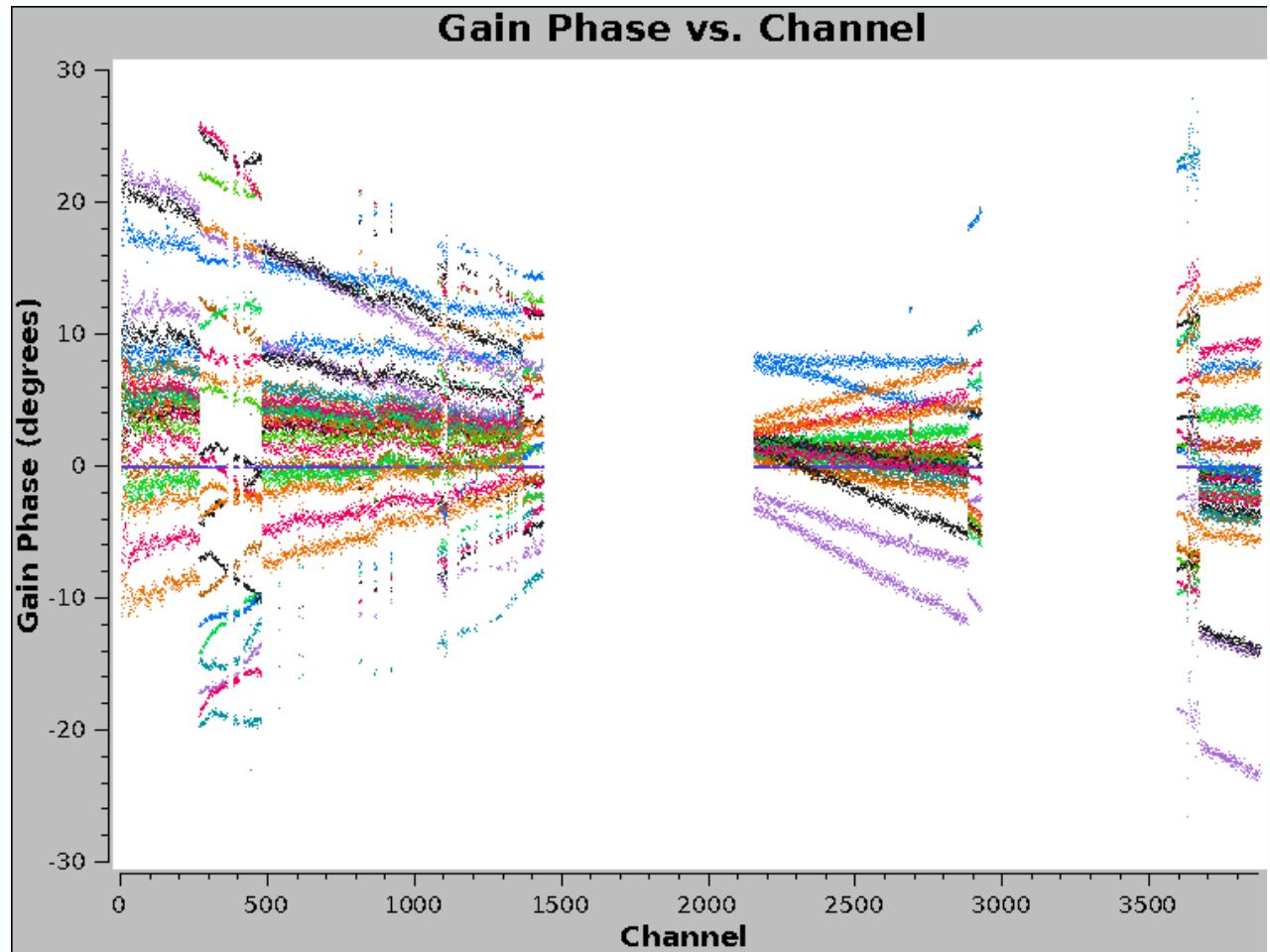
Initial QA: Calibration Solutions

- Bandpass solutions



Initial QA: Calibration Solutions

- Bandpass solutions
- And more...



A Good Framework

- Pipeline outputs calibrated MMS + MSs/MMSs split by field
- You insert your scripts at start, middle or end (e.g. WSClean)
- Each job/script is a logical step that does/doesn't use MPI, and optionally uses a different container
- HPC-friendly – dynamically uses resources & submits to queue
- Use cases we currently support
 - Full stokes calibration + Stokes I only calibration (minimal speedup)
 - Narrow band (spectral line) calibration, full-band calibration
 - Single MS (speedup for small BW), multi-MS
 - Inserting your own scripts (hard-coded or read config file)
- <https://idia-pipelines.github.io/docs/processMeerKAT>

Future Development

- Selfcal and AW Projection (see Krishna's talks)
- Optimisation of resources / performance
 - Currently takes ~1 day to process 64 dish 4k data (~2 TB)
 - Split data by intent during beginning, and simultaneously calibrate / flag
 - More dynamic use of threads and memory per script per intent (based on benchmarking)
 - Partitioning (IO), flagging (RAM), imaging (CPU)
 - Will see a significant speedup, necessary for arrival of 32k data
- Comprehensive data quality assessment



Inter-University Institute
for Data Intensive Astronomy

Data Quality Assessment



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

WESTERN SYDNEY
UNIVERSITY



Jordan Collier | 20 Feb 2019 | MIGHTEE workshop

Data Quality Assessment

- Already implemented: quality assessment of data products out of processing pipeline
 - i.e. scientific DQA at end of workflow
- End-to-end analysis of MeerKAT (continuum) image / catalogue
- Presents validation report
- Produces several tables summarising data and DQA metrics
- Runs in jupyter notebook on IDIA cloud with interactive plots

MeerKAT Continuum Data Validation Report							
Image							
File: 'DEEP_2_mfs.sc7.image.tt0.fits'							
Date	Field Centre	Central Frequency (MHz)	Synthesised Beam (arcsec)	Median r.m.s. (uJy)	Image peak (Jy)	Dynamic Range (image peak / worst r.m.s.)	Sky Area (deg ²)
2017-04-04T07:48:03.798888D:01:11	04:13:26.8223	1273.88	4.4 x 4.3	14	0.03	5E+02	6.27
Catalogue							
File: 'DEEP_2_mfs.sc7.image.tt0_aegean_comp.fits'							
Source Finder	Flux Type	Number of sources ($\geq 5.0\sigma$)	Multi-component islands	Sum of image flux vs. sum of catalogue flux	Median spectral index	Source Counts Xred ⁻²	Source Counts Xred ⁻²
aegean	integrated	2769	113	1.1 Jy vs. 1.4 Jy		191.10	
Cross-matches							
Survey	Frequency (MHz)	Cross-matches	Median offset (arcsec)	Median flux ratio	Median spectral index		
SUMSS	843.0	35	-0.63 ± 2.19 (RA) 0.03 ± 2.85 (Dec)	0.15 ± 0.14 (extrapolated)	-5.40 ± 2.46		
GLEAM	201.0	14	-5.72 ± 4.42 (RA) 6.78 ± 10.30 (Dec)	0.04 ± 0.03 (extrapolated)	-2.59 ± 0.52		
MeerKAT continuum validation metrics							
Flux Ratio (MeerKAT / SUMSS-extrapolated)	Flux Ratio Scatter (MeerKAT / SUMSS-extrapolated)	Positional Offset (arcsec) (MeerKAT – SUMSS)	Positional Offset Scatter (arcsec) (MeerKAT – SUMSS)	Resolved Fraction from int/peak Flux (MeerKAT)	Spectral Index (MeerKAT-SUMSS)	Source Counts Xred ⁻² (MeerKAT)	Source Counts Xred ⁻² (MeerKAT)
0.15	0.14	0.63	3.60	0.21	-5.40	191.10	

Data Quality Assessment

The screenshot shows a Jupyter Notebook interface running in a Chrome browser window. The title bar reads "Data Quality Assessment". The notebook content is as follows:

Data Quality Assessment

Input a radio continuum image and produce an external validation report (in html) of positions, fluxes, source counts, etc in the current directory.

Import classes and functions from script

```
In [ ]: from MeerKAT_quality_assessment import *
from IPython.display import display, HTML
%matplotlib notebook
```

Set input fits file and parameters

```
In [ ]: #input data
img = '/users/jcollier/DEEP/DEEP_2_mfs.sc7.image.tt0.fits'
noise = None
main_cat_config = None

#pyBDSF catalogue
# noise = '../DEEP_2_mfs.sc7.image.tt0_pybdsm/run1_17Nov2017_18.19.48/background/DEEP_2_mfs.sc7.image.tt0.pybdsm.rmsd_I'
# main_cat_config = 'deep_config.txt'

#script config variables
WD = '/users/jcollier/DEEP/'
main_dir = '/users/jcollier/QAfiles/'
refind = False
redo = True
aegean_params = ''
ncores = 8

#output/display variables
verbose = True
```

Data Quality Assessment

- Have started looking at DQA of early pipeline data products
- In close collaboration with MeerKAT SDP.
 - Framework to measure quality of pipelines.
 - Standard set of metrics between all pipelines
 - Defer processing to most efficient pipelines?
 - Mapping science requirements from LSPs to technical requirements for pipelines

Data Quality Assessment

Table 1: Selected requirements and specifications from various MeerKAT imaging LSPs

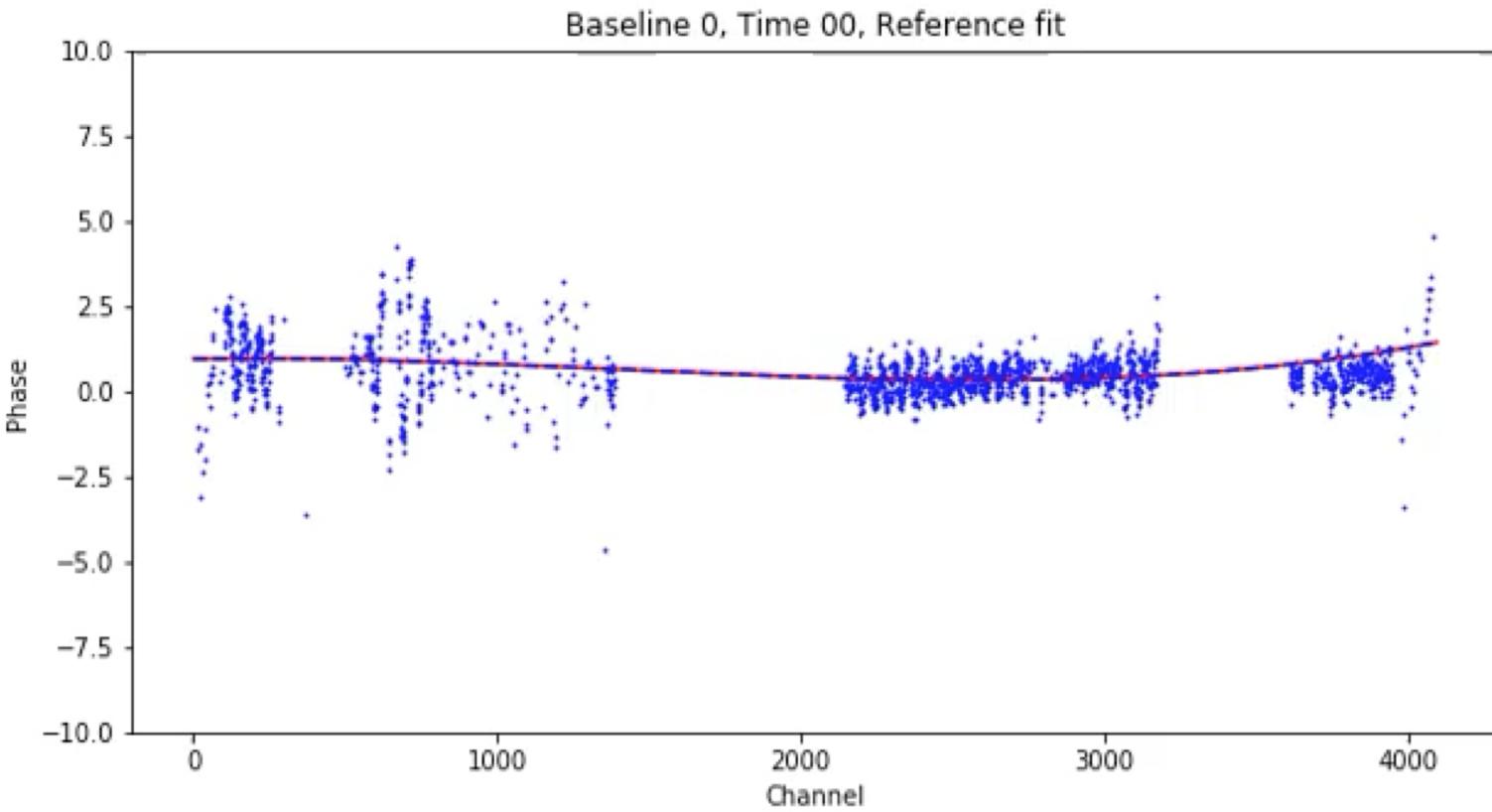
LSP	Sensitivity ($\mu\text{Jy beam}^{-1}$)	Dynamic Range	Velocity Resolution (km s^{-1})	Redshift Range	Area (deg 2)	RM precision (rad m $^{-2}$)	Column Density Sensitivity	H I Mass (M $_{\odot}$)
MIGHTEE (L-band)	2 ^a / 90 ^b	$\geq 10^5$ ^c	6 ^d	$z \lesssim 0.5$	20	~ 1	$\sim 1 \text{ M}_{\odot} \text{ pc}^{-2}$ ^e	$\gtrsim 2 \times 10^{9}$ ^f
MIGHTEE (S-band)	1 ^g	$\geq 10^5$ ^c	-	$z \lesssim 0.5$	5.5	~ 1	-	-
MIGHTEE (UHF-band)	6 ^a	$\geq 10^5$ ^c	-	$z \lesssim 0.5$	3.5	~ 1	-	-
LADUMA (L-band)	45 ^b	-	6 ^d	$0 \leq z \leq 0.58$	0.9–2.2	-	-	$\sim 10^{7.5-10.5}$
LADUMA (UHF-band)	26 ^h	-	8 ⁱ	$0.42 \leq z \leq 1.45$	1.8–5.4	-	-	$\sim 10^{9.2-10.5}$
FORNAX	100 ^j	-	~ 1	$z \sim 0$	~ 12	-	$\sim 0.1^k - 5^l \times 10^{19} \text{ cm}^{-2}$	$\gtrsim 5 \times 10^5$ ^l
MHONGOOSE	74 ^j	-	16	$z \sim 0$	$> 45^m$	$\lesssim 1$	$0.55^n - 7.5^o \times 10^{18} \text{ cm}^{-2}$	$\sim 10^{6-11}$
MALS (L-band)	3 / 500 ^p	-	~ 5	$0 < z < 0.85^q$	1000 ^r	-	$> 10^{19} \text{ cm}^{-2}$	$\sim 5 \times 10^4$
MALS (UHF-band)	3 / 600 ^p	-	~ 5	$0.4 < z < 1.87^q$	700 ^r	-	$> 10^{19} \text{ cm}^{-2}$	$\sim 5 \times 10^4$
THUNDERKAT	$\sim 1-1000^s$	-	-	-	-	-	-	-

Data Quality Assessment

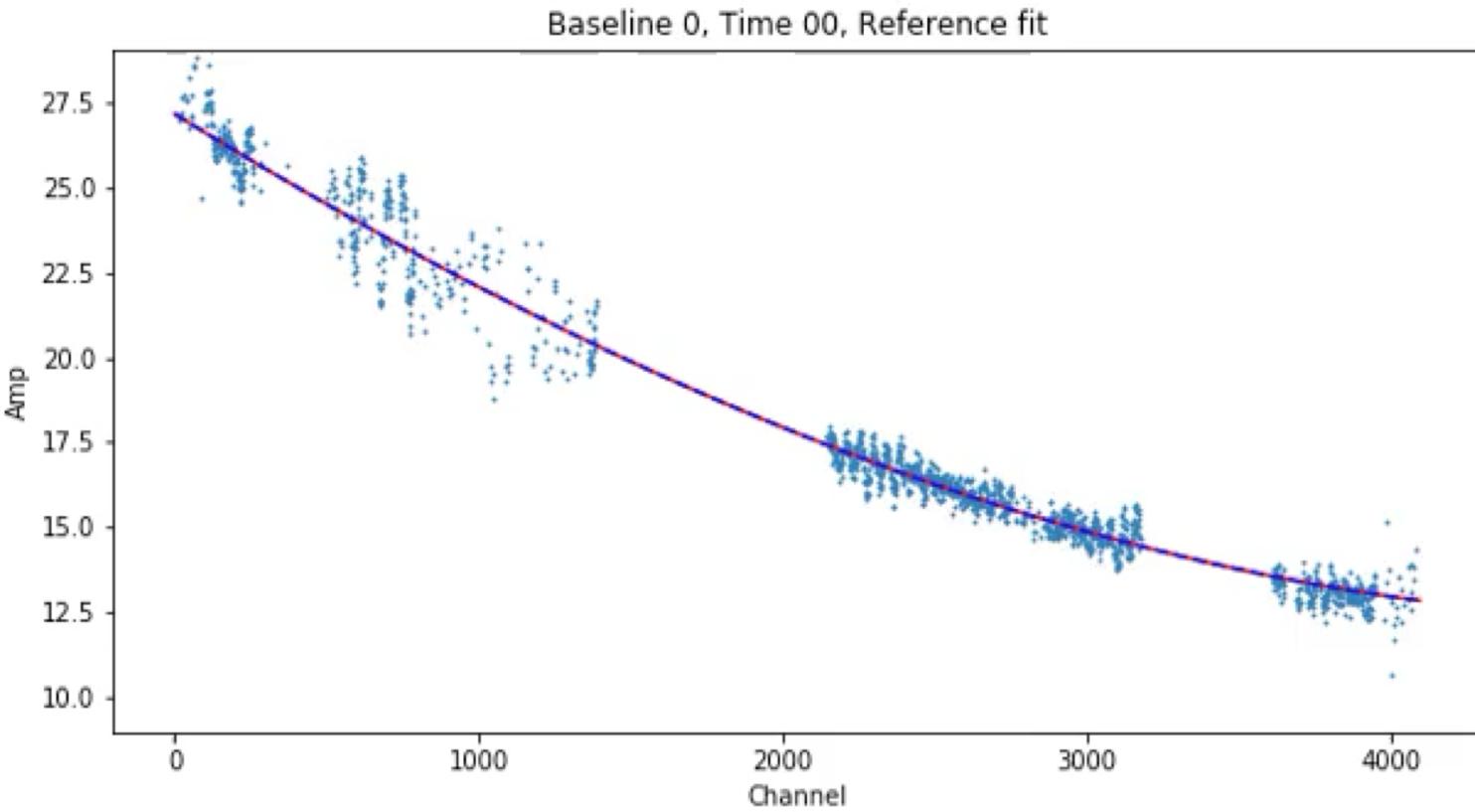
Table 2: Data quality metrics for a simple calibration pipeline. This table is not exhaustive, but represents a selection of metrics we have drafted.

Step	Distribution / plot	Statistic / Metric	Computation	Tolerance
Bandpass calibration	Calibrated bandpass (amplitude and phase) as a function of frequency (per timestamp)	Residual of polynomial fit	Mean reduced χ^2 squared of polynomial fit to each timestamp, compared to reference timestamp ^a	< 10
	Calibrated bandpass (amplitude and phase) as a function of frequency	Normalised median absolute deviation	Maximum normalised absolute deviation from polynomial fit	< 5
Bandpass flagging	Calibrated amplitude as a function of frequency	Fraction of channels flagged	Number of channels with >50% of visibilities flagged divided by total number of channels	< 10%
Phase calibration	Complex gain solutions as a function of time	Outlier metric	Running median?	?
	Calibrated amplitude as a function of frequency	Normalised median absolute deviation	Normalised median absolute deviation from polynomial fit	< 5
Phase cal. flagging	Calibrated amplitude as a function of time	Fraction of timestamps flagged	Number of timestamps with >50% of visibilities flagged divided by total number of timestamps	< 10%
Target flagging	Amplitude as a function of frequency	Fraction of data flagged	Fraction of visibilities flagged divided by total visibilities	< 20%
	Amplitude as a function of frequency	Normalised median absolute deviation	Normalised median absolute deviation	< 5

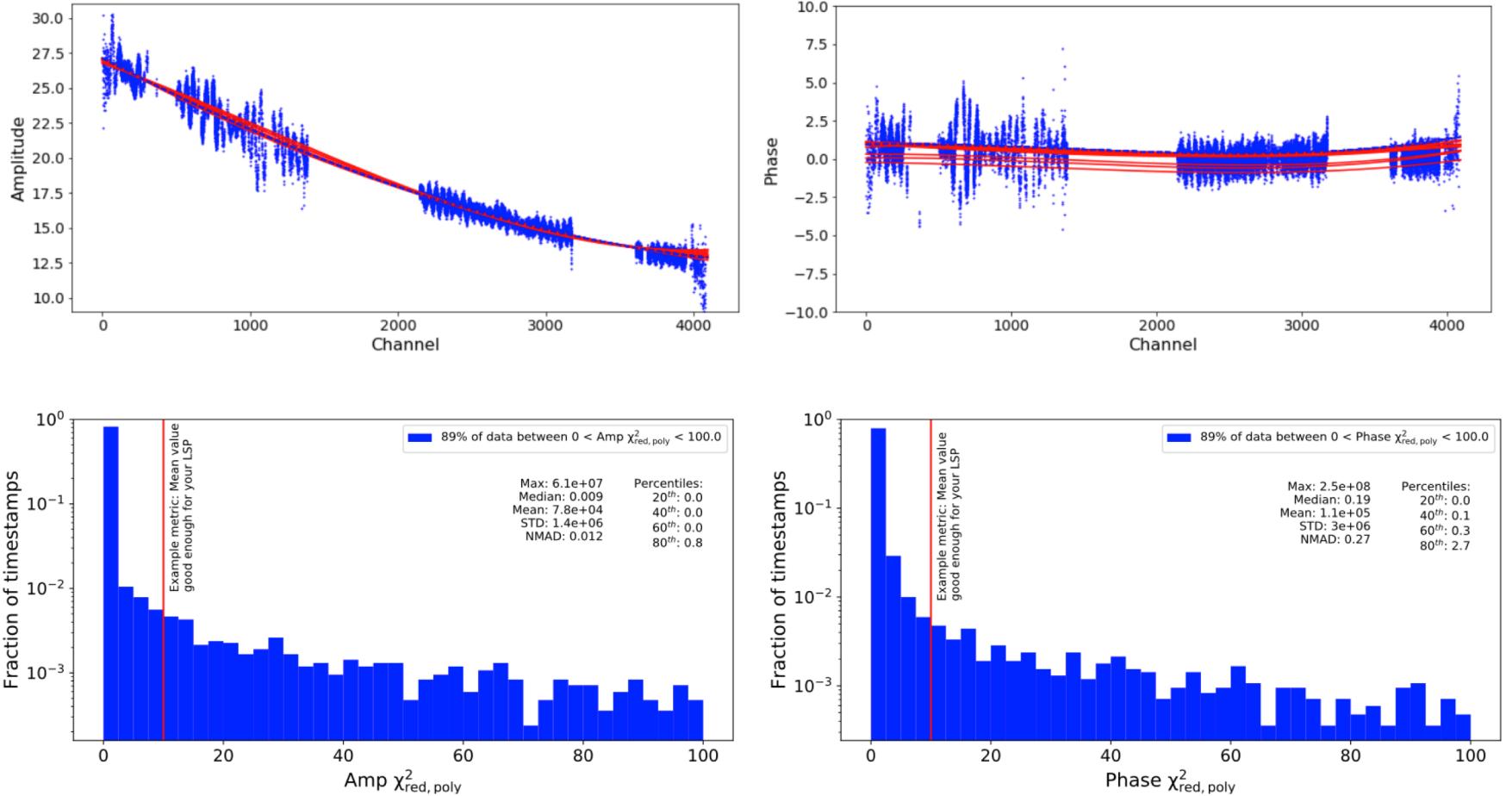
Data Quality Assessment



Data Quality Assessment



Data Quality Assessment



Data Quality Assessment

- DQA being discussed amongst SKA pathfinders in general, within SPARCS DQA WG
 - <http://spacs.pbworks.com/w/page/126067640/dataquality>
- Drafted list of metrics between us
 - https://docs.google.com/spreadsheets/d/1l8xOdoPW6LhoWZMLeB5W4tcwzdgw18FWdh5DxnK6_vs/edit#gid=0

Summary

- The IDIA “processMeerKAT” pipeline is an efficient, user-friendly pipeline, that is widely tested and documented
- It runs on the Ilifu cluster, making dynamic use of resources, and containers, and presents a good framework for pipelines
- Many use cases are supported, incl. inserting your own scripts
- Even the quick look images are good
- Coming soon: selfcal, AW projection, optimisation/speedup

THANK YOU

Dr Jordan Collier

Iifu Support Astronomer, IDIA
Department of Astronomy,
University of Cape Town

Adjunct Fellow, Western Sydney University
School of Computing, Engineering and
Mathematics

Jordan@idia.ac.za
+27 664 343 953 (RSA Mobile)
+61 414 443 622 (AU Mobile / WhatsApp)



Inter-University Institute
for Data Intensive Astronomy



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

