The Very Large Array Data Processing Pipeline

Brian R. Kent[1], Joseph S. Masters[1], Claire J. Chandler[2], Lindsey E. Davis[2], Jeffrey S. Kern[1], Juergen Ott[2], Frank K. Schinzel[2], Drew Medlin[2],
Dirk Muders[3], Stewart Williams[4], Vincent C. Geers[4], Emmanuel Momjian[2], Bryan J. Butler[2], Takeshi Nakazato[5], Kanako Sugimoto[5]

Institutions: (1) NRAO, Charlottesville, VA (2) NRAO, Socorro, NM (3) Max-Planck-Institut für Radioastronomie, Bonn, Germany
(4) UK ATC, Edinburgh, United Kingdom (5) NAOJ, Tokyo, Japan

Abstract

We present the VLA Pipeline, software that is part of the larger pipeline processing framework used for the Karl G. Jansky Very Large Array (VLA), and Atacama Large Millimeter/sub-millimeter Array (ALMA) for both interferometric and single dish observations.

Through a collection of base code jointly used by the VLA and ALMA, the pipeline builds a hierarchy of classes to execute individual atomic pipeline tasks within the Common Astronomy Software Applications (CASA) package. Each pipeline task contains heuristics designed by the team to actively decide the best processing path and execution parameters for calibration and imaging. The pipeline code is developed and written in Python and uses a "context" structure for tracking the heuristic decisions and processing results. The pipeline "weblog" acts as the user interface in verifying the quality assurance of each calibration and imaging stage. The majority of VLA scheduling blocks above 1 GHz are now processed with the standard continuum recipe of the pipeline and offer a calibrated measurement set as a basic data product to observatory users. In addition, the pipeline is used for processing data from the VLA Sky Survey (VLASS), a seven year community-driven endeavor started in September 2017 to survey the entire sky down to a declination of -40 degrees at S-band (2-4 GHz). This 5500 hour next-generation large radio survey will explore the time and spectral domains, relying on pipeline processing to generate calibrated measurement sets, polarimetry, and imaging data products that are available to the astronomical community with no proprietary period. Here we present an overview of the pipeline design philosophy, heuristics, and calibration and imaging results produced by the pipeline. Future development will include the testing of spectral line recipes, low signal-to-noise heuristics, and serving as a testing platform for science ready data products.

The pipeline is developed as part of the CASA software package by an international consortium of scientists and software developers based at the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), and the National Astronomical Observatory of Japan (NAOJ).

Design and Features

The pipeline consists of an execution framework written in Python. Its design is based around tasks, each with their own unit of functionality. Pipeline tasks are collected into serialized procedures and executed through a pipeline processing request XML file (PPR-XML) or Python CASA script. As each stage is completed through the pipeline execution, a result object is generated with all associated information – task inputs, heuristic decisions, computations, and quality assurance (QA) metrics. These result objects are collected in the pipeline context, and updated as each subsequent stage is completed, maintaining the current state of the pipeline and allowing for breakpoints and restarts.

The pipeline weblog acts as the user interface in understanding the results of the pipeline. Written as a responsive webpage with with Bootstrap and Python Mako templates, a user can diagnose any issues that may be present in the observation. All relevant metadata concerning the observations, data reduction notifications, QA metrics, and heuristic outcomes are displayed for each task. The weblog is designed to assist in gauging the quality of calibration and imaging, so that a user may make the appropriate decisions with any additional data reduction requirements or additional reprocessing (for example, additional flagging or evaluating data for future observations).

The pipeline software is designed around a package hierarchy designated by prefixes, including *h* (heuristics), *hif* (heuristics interferometry), *hifa* (ALMA), *hifv* (VLA), and *hsd* (single dish). For example, interferometric heuristics are designed with general Python classes and properties – VLA and ALMA inherit from these classes and further refine and override class methods and parameters that fit the particular telescope facility.

Usage and Application

The standard recipe used with most VLA scheduling blocks was developed for mid to high frequency continuum observations (*f* > 1 GHz). Reprocessing options are available for spectral line as well, allowing the user to manually identify continuum regions in a spectral window. Scan intents of CALIBRATE_FLUX, AMPLI, and PHASE are required, as is a signal-to-noise of ~3 for each spectral window of a calibrator per integration.

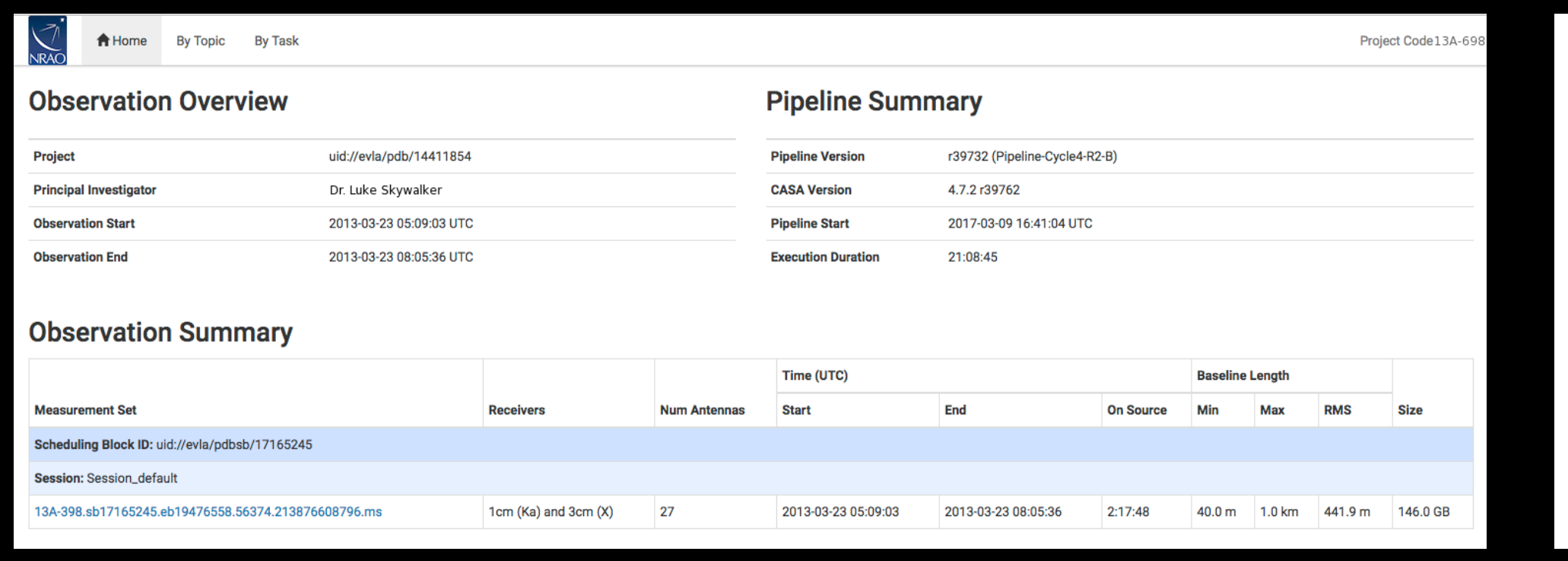
VLA tasks are usually designated by *hifv*. For imaging tasks that are shared with ALMA, the more generic *hif* is used. Some tasks are functionally required for pipeline operation, including import and application of the calibration tables. Nevertheless, the tasks are designed to be atomic and flexible such that some can be removed depending on the data being reduced and/or time constraints. Some pipeline runs may wish to omit Hanning smoothing or calibrator imaging. Others may wish to forgo some of the diagnostic summary plots.

In addition to the measurement set with calibrated visibilities, the pipeline produces a collection of standard products with the exportdata task. The products include the weblog, CASA text execution logs, pipeline context, applied calibration tables, and a Python script record of tasks and keyword arguments to reproduce the results. A restoration script that can be used to restore the calibration state without rerunning the entire pipeline is also created.

An example Python CASA pipeline execution script is shown below:

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'observatory', 'Karl G. Jansky Very Large Array')
context.set_state('ProjectSummary', 'telescope', 'VLA')
try:
    hifv_importdata(vis=[’15B-342.sdm’])
    hifv_hanning(pipelinemode=‘automatic’)
    hifv_flagdata(tbuff=0.0, scan=True, hm_tbuff='1.5int')
    hifv_vlasetjy(fluxdensity=-1, scalebychan=True, spix=0, reffreq='1GHz')
    hifv_priorcals(tecmaps=False)
    hifv_testBPdcals(weakbp=False)
    hifv_flagbaddef(pipelinemode=‘automatic’)
    hifv_checkflag(pipelinemode=‘automatic’)
    hifv_semiFinalBPdcals(weakbp=False)
    hifv_checkflag(checkflagmode='semi')
    hifv_semiFinalBPdcals(weakbp=False)
    hifv_solint(pipelinemode=‘automatic’)
    hifv_fluxboot(pipelinemode=‘automatic’)
    hifv_finalcals(weakbp=False)
    hifv_applycals(flagdetailedsum=True, flagsum=True)
    hifv_targetflag(intents='*CALIBRATE*,*TARGET*')
    hifv_statwt(pipelinemode=‘automatic’)
    hifv_plotsummary(pipelinemode=‘automatic’)
    hif_makeimlist(specmode='cont', nchan=-1, calmaxpix=300, intent='PHASE,BANDPASS')
    hif_makeimages(hm_masking=‘none’)
    hifv_exportdata(pipelinemode=‘automatic’)
finally:
    h_save()
```

The weblog example below shows sample tables and plots that summarize the observational metadata, state of pipeline completion and runtime, observing fields and intents, relevant weather information, as well as spatial, spectral, and scan configurations.



Observation Overview

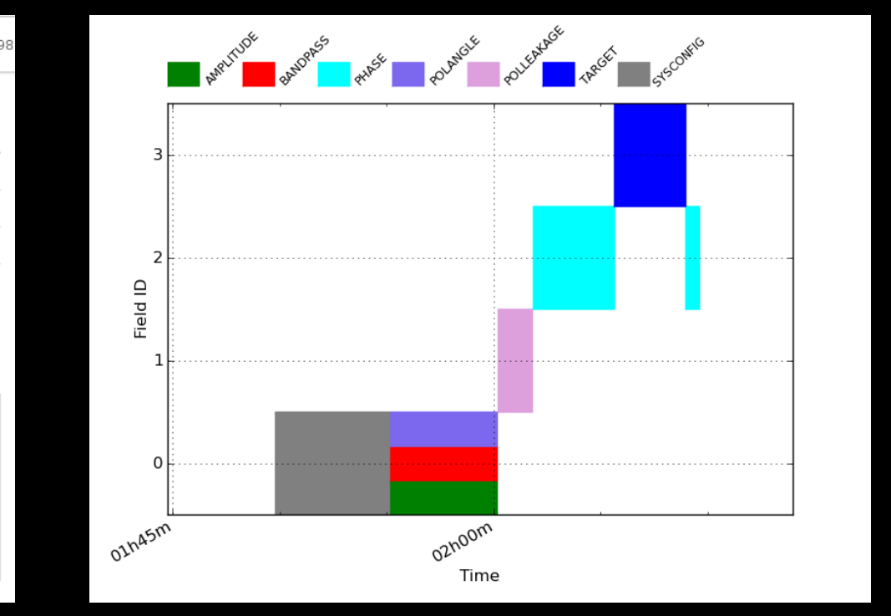
Project	uid://evla/pdb/14411854
Principal Investigator	Dr. Luke Skywalker
Observation Start	2013-03-23 05:09:03 UTC
Observation End	2013-03-23 08:05:36 UTC

Pipeline Summary

Pipeline Version	r39732 (Pipeline-Cycle4-R2-B)
CASA Version	4.7.2 r39762
Pipeline Start	2017-03-09 16:41:04 UTC
Execution Duration	21:08:45

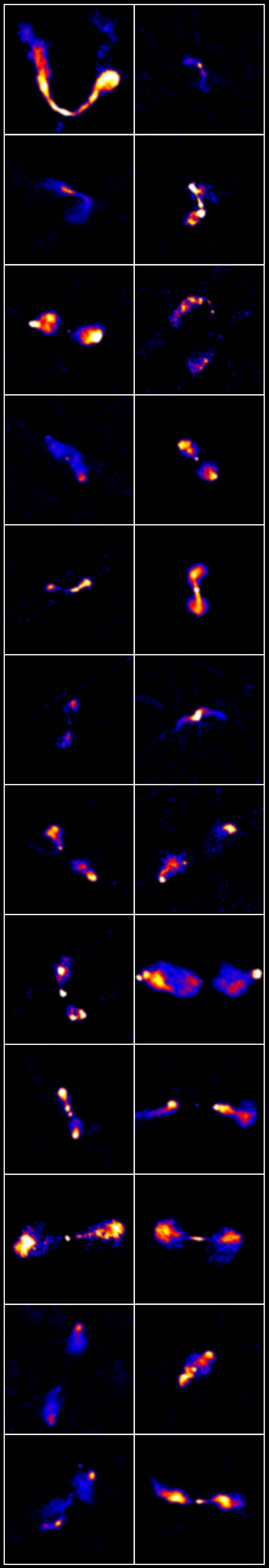
Observation Summary

Measurement Set	Receivers	Num Antennas	Time (UTC)			Baseline Length			Size
			Start	End	On Source	Min	Max	RMS	
Scheduling Block ID: uid://evla/pdbsb/17165245									
Session: Session_default									
13A-398.sb17165245.eb19476558.56374.213876608796.ms	1cm (Ka) and 3cm (X)	27	2013-03-23 05:09:03	2013-03-23 08:05:36	2:17:48	40.0 m	1.0 km	441.9 m	146.0 GB



A thorough and detailed examination of using the pipeline for VLA data reduction has been created on the CASA Guides website at:

https://casaguides.nrao.edu/index.php/VLA_CASA_Pipeline-CASA4.7.2



Calibration

The standard VLA continuum data reduction procedure determines, on the basis of a priori factors and from observations of standard calibration sources, the corrections to the raw data amplitude, phase, and visibility weights to be applied to the data. This process also determines the flags that are needed to remove bad data due to instrumental faults, RFI, bad antennas, and other causes of error. This process only includes the derivation of the complex gain and bandpass calibration factors known through previous measurements or determined by the observations of calibrators and transferred to the observations of the target observations.

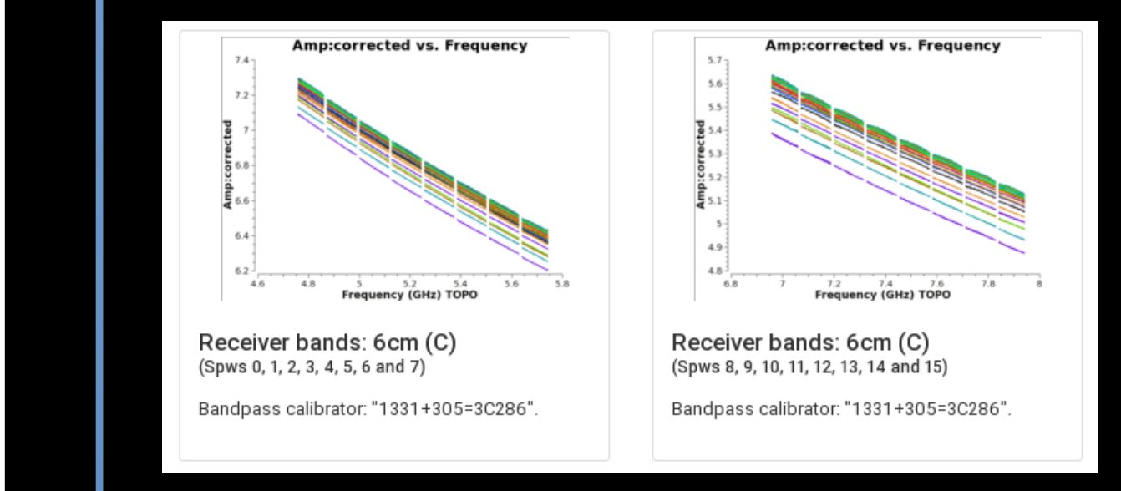
Calibration stages consist of:

1. Import from SDM to measurement set and application of initial online flags (off-source, focus error, subreflector error)
2. Determination and application of derived flags (RFI, bad antennas, shadowing, other)
3. Switched power amplitude calibration and antenna gain curves
4. Flux scale calibration (using standard sources)
5. Complex Delay and Bandpass Calibration, Complex Gain Calibration, followed by additional heuristic flagging
6. Flux density bootstrapping (from primary to secondary calibrators)
7. Interpolation and Application of Cumulative Calibration
8. Final Flagging of Data (insufficient or failed calibration, RFI) and statistical weighting of visibilities
9. Diagnostic calibrator imaging
10. Output of Quality Assurance (QA) information, plots, images, caltables, logs, and execution scripts

In addition, the standard VLASS calibration procedure includes different steps for heuristic flagging with RFLAG and TFCROP, as well as polarization calibration.

See the VLA science website for information on executing the pipeline:

https://science.nrao.edu/facilities/vla/data-processing/pipeline/



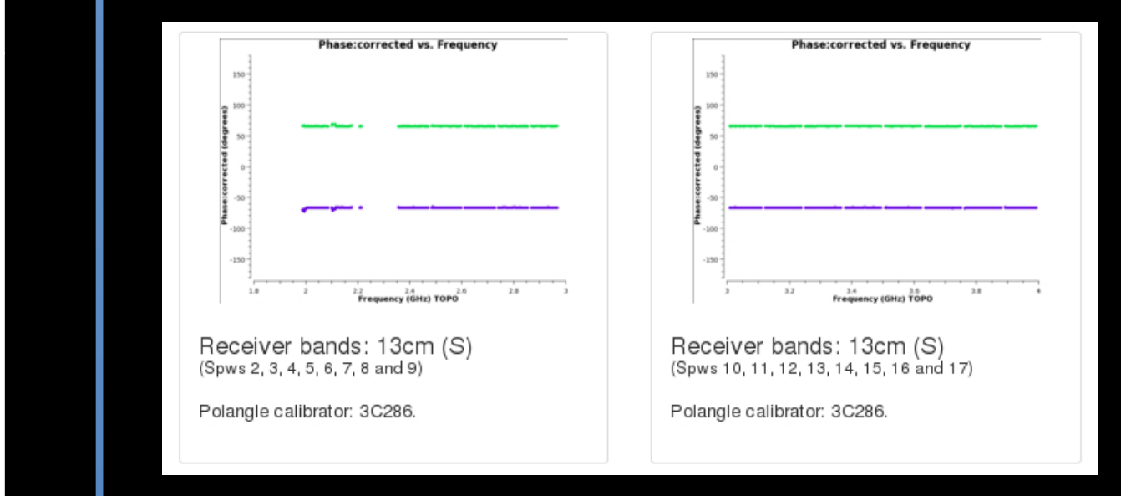
Receiver bands: 6cm (C)
(Spws 0, 1, 2, 3, 4, 5, 6 and 7)

Bandpass calibrator: "1331+305=3C286".

Receiver bands: 6cm (C)
(Spws 8, 9, 10, 11, 12, 13, 14 and 15)

Bandpass calibrator: "1331+305=3C286".

The two figures at left show diagnostic QA plots from the VLA plot summary page, showing the corrected amplitude vs. frequency for the bandpass calibrator 3C286 at C-band. Plots are separated by baseband, each with eight color coded spectral windows.



Receiver bands: 13cm (S)
(Spws 2, 3, 4, 5, 6, 7, 8 and 9)

Polangle calibrator: 3C286.

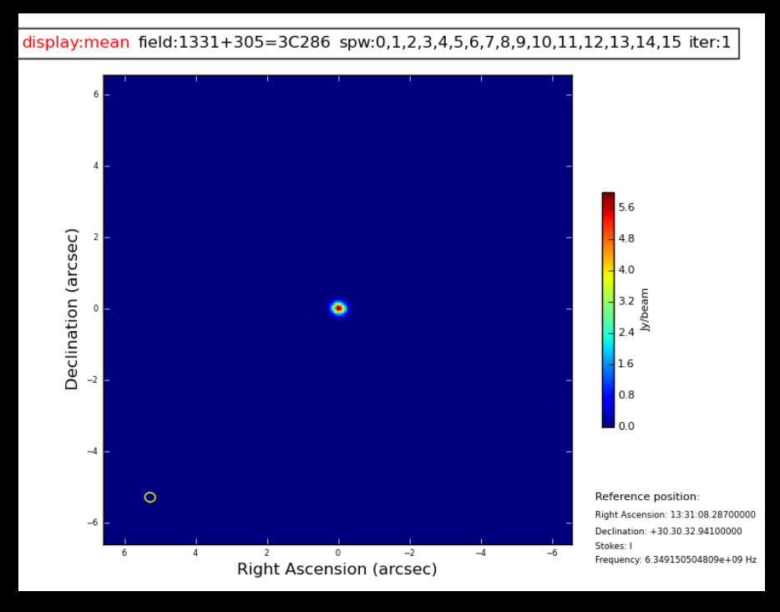
Receiver bands: 13cm (S)
(Spws 10, 11, 12, 13, 14, 15, 16 and 17)

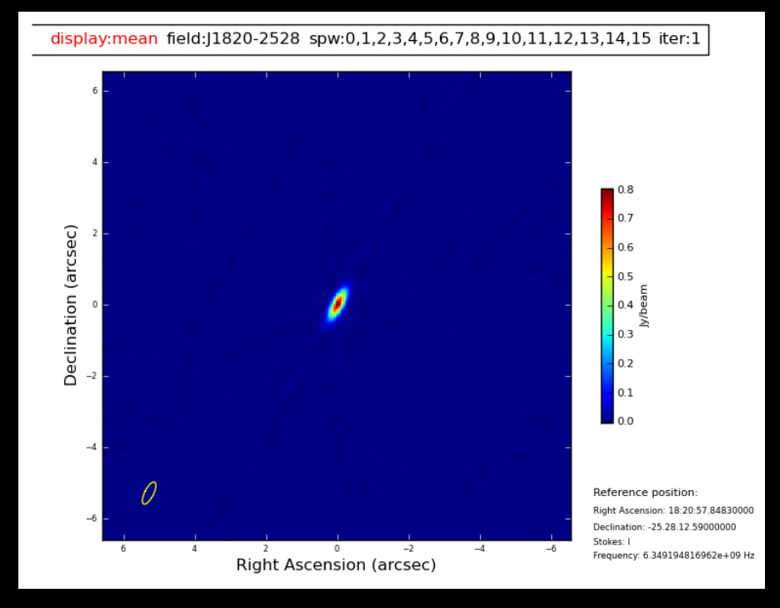
Polangle calibrator: 3C286.

The two figures at left show diagnostic QA plots from the VLASS polarization calibration summary, showing the corrected phase vs. frequency for the polarization angle calibrator 3C286 at S-band. Plots are separated by baseband, color coded by correlations LR and RL. A single spectral window has been flagged due to RFI in this particular scheduling block.

Imaging

Imaging in the standard VLA reduction recipe currently images all calibrators for all spectral windows. Bandpass calibrator 3C286 and phase calibrator J1820-2528 have been imaged in the figures at right. The cleaned C-band images cover a bandwidth of 2 GHz, and are only examples produced with the standard calibration pipeline – no target images are produced yet for regular VLA scheduling blocks. Options are available as keyword arguments to image per spectral window or across the entire band.





The Very Large Array Sky Survey	
Frequency	2-4GHz
Resolution	2.5 arcsec
Sky coverage	All sky north of Dec > -40 deg (33885 sq. deg.)
Sensitivity per epoch	Goal: 120 microJy RMS
Combined (3 epoch sensitivity)	Goal: 69 microJy RMS
Polarization	I,Q,U
Cadence	3 epochs separated by 32 months
Start Date	September 7, 2017
Expected source count	~10 million

The VLA Sky Survey (VLASS) is currently being used a test case for science target imaging with its quick look pipeline. As VLASS is a focused S-band all-sky survey with fixed observation parameters, the imaging heuristics are somewhat less varied than considering all possible VLA observation configurations. The table above details some of the basic survey parameter goals that will be obtained. VLASS imaging will consist of:

- Quick Look (QL) imaging triggered after every scheduling block is observed (e.g. for transient identification)
- Per-epoch, higher quality imaging triggered after the last observation each configuration
- Cumulative imaging triggered after each epoch beyond the first, incorporating all previous data

The central column of figures at left show preliminary 2 arcminute images automatically generated with the quick look imaging pipeline. Each image is scaled with a logarithm transfer function, and shows a sample of the ***previously unresolved objects*** that are being detected with VLASS.

Learn more about VLASS, its public data, and pipeline imaging at: https://science.nrao.edu/science/surveys/vlass

Science Ready Data Products

Part of the mission of NRAO is to provide robust data products to the astronomical community. VLASS, ALMA, and the cooperative effort of pipeline development is serving as a pathfinder toward exciting new facilities like the Next Generation Very Large Array project.

The pipeline is in use every day, and future development will be driven by the need for producing science ready data products for the scientific community. It continues to leverage the data output of VLASS to improve and refine the heuristics for all VLA observations. The knowledge gained from the thousands of pipeline execution runs will be built into an automated quality assurance (QA) system for the VLA pipeline weblog.

Visit the poster 342.13 by J. Kern, *Science Ready Data Products and the ngVLA*, for more information.

References

Davis, L., Williams, S., Nakazato, T., Lightfoot, J., Muders, D., Kent, B. 2015, ADASS XXIV, ASP Vol. 495, 301

Chandler, C., Myers, S., Lacy, M, Hiriart, R., McLaughlin, C. 2017, VLASS Technical Specifications, NRAO-305-258

Abdalla, F. et al. 2014, The Jansky Very Large Array Survey, VLASS Science Group
https://safe.nrao.edu/wiki/pub/JVLA/VLASS/VLASS_final.pdf

Bootstrap: https://getbootstrap.com/

Mako: http://www.makotemplates.org/