NRAO An NSF Facility

Pipelines!

National Radio Astronomy Observatory

Loránt Sjouwerman

Atacama Large Millimeter/submillimeter Array
Karl G. Jansky Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

Associated Universities, Inc.

NSF

NRAO

Aim

- Radio interferometry data sets are becoming complex and huge, and increasingly cumbersome to deal with
- Pipelines are already being widely used to process (radio interferometry) data
- Pipelines save a lot of time, but each have their pro's and con's

Overview

- Pipelines (data reduction),
  - what and why?
  - principles!
  - what's out there?
- CASA, AIPS, other pipelines
  - Example: ALMA pipeline by Liz Humphreys hereafter!
- Design your own!
- Cautionary notes

Pipeline?

- Dictionary (noun)
  - 1] A long pipe, typically underground, for conveying oil, gas, etc. over long distances
    - 1.1] A channel or system supplying goods or information
  - 2] Computing: A linear sequence of specialized modules used for "processing"
  - 3] (In surfing) the hollow formed by the breaking of a very large wave

NRAO

Pipeline?

- Wikipedia.org (e.g.)
  - Pipeline (computing), a chain of data-processing stages and/or processes

A pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one

Pipeline?

- Observatories (support part):
  - Telescope data describing an observation, processed from its original state (…) to the form in which it is presented to the end user
  - Here: "raw"/archive data → observer

Pipeline?

- Observatory directors:
  - Proposal →
    (observer/staff → telescope → staff)
    → "science ready images"
  - What are "science ready images"?
  - What about the resources?

Pipeline?

- Pipelines are useful to process data in a consistent way, for procedures that are known or can be derived automatically
- **Heuristics!**

Pipeline?

- Pipelines have a defined input or starting point, and a purpose or goal with a defined final data product or outcome

- Not always explicit...

Pipeline?

- A "pipeline" can be a simple few-line script, to a multi-purpose, flexible and complex data processing path with many different procedures, even a pipeline of pipelines, all depending on the goal

NRAO

Why pipelines?

- Observatory perspective

- Observations → publications!
  - Telescopes are more and more complex
  - Can collect a wealth of data
- Data reduction → results, a bottle neck?
  - Speed up or take away bumps
- Happy scientific community
  - Faster turn-around of effort
  - Ease of use → more/new observers

Why pipelines?

- Observers perspective

- Observations → publications!
  - Telescopes are more and more complex
  - Can collect a wealth of data
- Data reduction → results, a bottle neck!
  - Speed up or take away bumps
- Happy scientific community
  - Faster turn-around of effort
  - Ease of use → more/new observations

NRAO

Why pipelines?

- Also (observer perspective):
  - Observatory generally best knows, e.g.,
    - What to flag as bad data
    - How to optimize calibration
  - Observatory has the infrastructure
  - Pipeline can reduce data set size
  - Allows for science instead of processing
- But note:
  - Science/result is domain of observer!

Why pipelines?

- **Pipelines are good!**
  - Provide consistent approach
  - Apply accumulated knowledge
  - Saves distraction from doing science
- However, every pipeline has its limitations
  - Make sure these are understood
  - Best effort approach

Principles of pipelines

- Pipelines do what they are programmed to do and typically do it well and efficiently:
  - take input, attempt goal, deliver product
- Ideally only use the information contained in the data, possibly with hooks for override
  - Hooks should be for fine-tuning only, not for requiring human intervention to run

Principles of pipelines

- Apply sequence of processes, each tuned with its own heuristics to achieve sub-goal
- Each sub-goal should be evaluated (QA) before attempting next step:
  → go/no-go
- Reporting is one of the key final products
  (e.g., log-file, progress plot, process timing)

Principles of pipelines

- Main types of pipelines:
  - **Observatory** run pipelines
    - Push data to community
  - **User** run pipelines
    - (re)Process project
  - **Private**/personal pipelines…
    - Any particular goal

Principles of pipelines

- Observatory pipelines are run by staff, run on all eligible observations with a general goal and typically not well advertised or documented
- User pipelines are run by the user on a limited number of observations with a specific goal in a data reduction package and therefore usually better documented

NRAO

Principles of pipelines

- For example:
  - extract observation from archive
  - load into data reduction package
  - remove/flag bad data
  - calibrate instrument (dly,bp,pol,flx)
  - calibrate environment (d/dt)
  - apply total calibration
  - make images
  - deliver products to final destination

  each step/sub-goal has its heuristics/QA

Principles of pipelines

- Quality of processing and QA more important than processing speed (24/7, future hardware upgrades)

- Partial run and/or (re)run with additional flagging, parameter override and other flexibility from a file or by hand should be built in from the start, not by many (hacked) versions of the code

NRAO

Principles of pipelines

- Pipelines can do all the nuisance and administration, be fine-tuned for the instrument and accumulated knowledge (heuristics), independent of science goal
  - great for calibration
- Science goal for the imaging artwork (i.e., "science ready") typically needs some extra information not known to the pipeline
  - limited to "reference images"

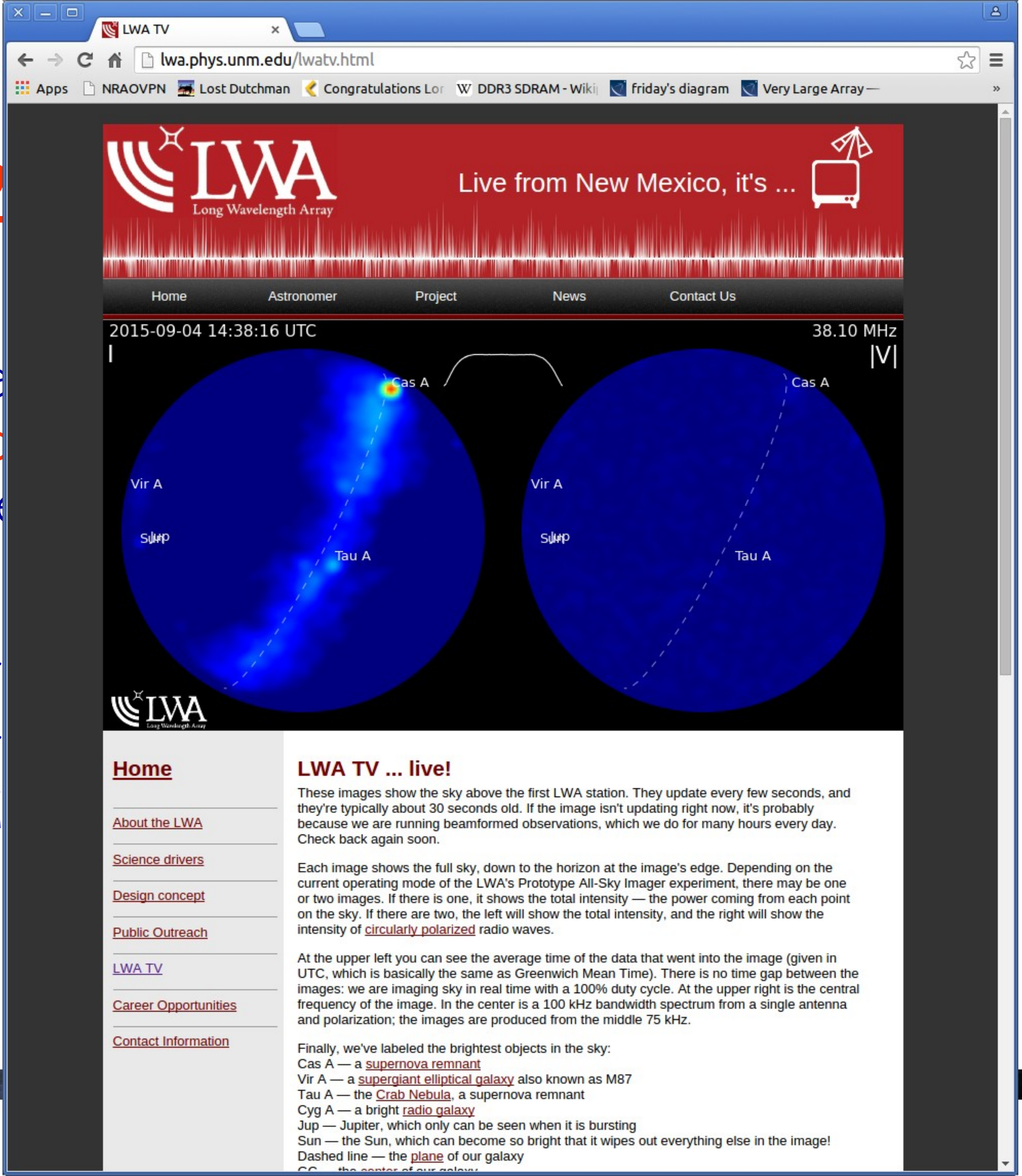
NRAO

Pipelines in the field

- Many observatories have pipelines for a specialized observing mode and a specific deliverable for non-user projects
- Examples:
  - Sky surveys like NVSS
  - Sky monitors like LWA-TV
  - Piggy-back observations like VLITE
- Code typically not publicly available

Pip

- Many obse
  specialize
  deliverabl
- Examples:
  - Sky
  - Sky
  - Pig
- Code typi



Pipelines in the field

- Many observatories have pipelines for a specialized observing mode and a specific deliverable for non-user projects
- Examples:
  - Sky surveys like NVSS
  - Sky monitors like LWA-TV
  - Piggy-back observations like VLITE
- Code typically not publicly available

Pipelines in the field

- Observatory based pipelines for user data need to be more general due to the variety of observing projects, modes and strategies, and typically attempt flagging, calibration and "reference images"
- Examples:
  - EVN Calibration pipeline
  - VLA CASA scripted pipeline
  - ALMA pipeline
- Code, documentation "available" (support?)

NRAO

- Observ
  need
  variety
  strate
  calibra
- Examp
  -
  -
  -
- Code,

The EVN Pipeline

Stephen Bourke, 27 Aug 2010

The EVN pipeline was originally developed by Cormac Reynolds, it is currently being developed by Stephen Bourke.

The main pipeline script is evn.py. It uses three Python modules:

- evn_aips_tasks.py: Shortcut functions for executing AIPS tasks with most parameters defaulted to sensible values for the EVN. Parameters which must be set are mentioned in the docstring at the beginning, e.g., try

```
>>> help(evn_aips_tasks.runlwpla)
```

- evn_funcs.py: Other functions used by evn.py. Mostly provide simplified interaction with functions in evn_aips_tasks.py .
- crfuncs.py: Generic functions used by evn.py. At the moment this is just a couple of filehandling classes to allow better logging of what the pipeline is doing.

The pipeline is driven by an input file. An example can be found here: template.txt

If you want to grab it all in one go, here's a gzipped tar: pypeline-4.7.tar.gz

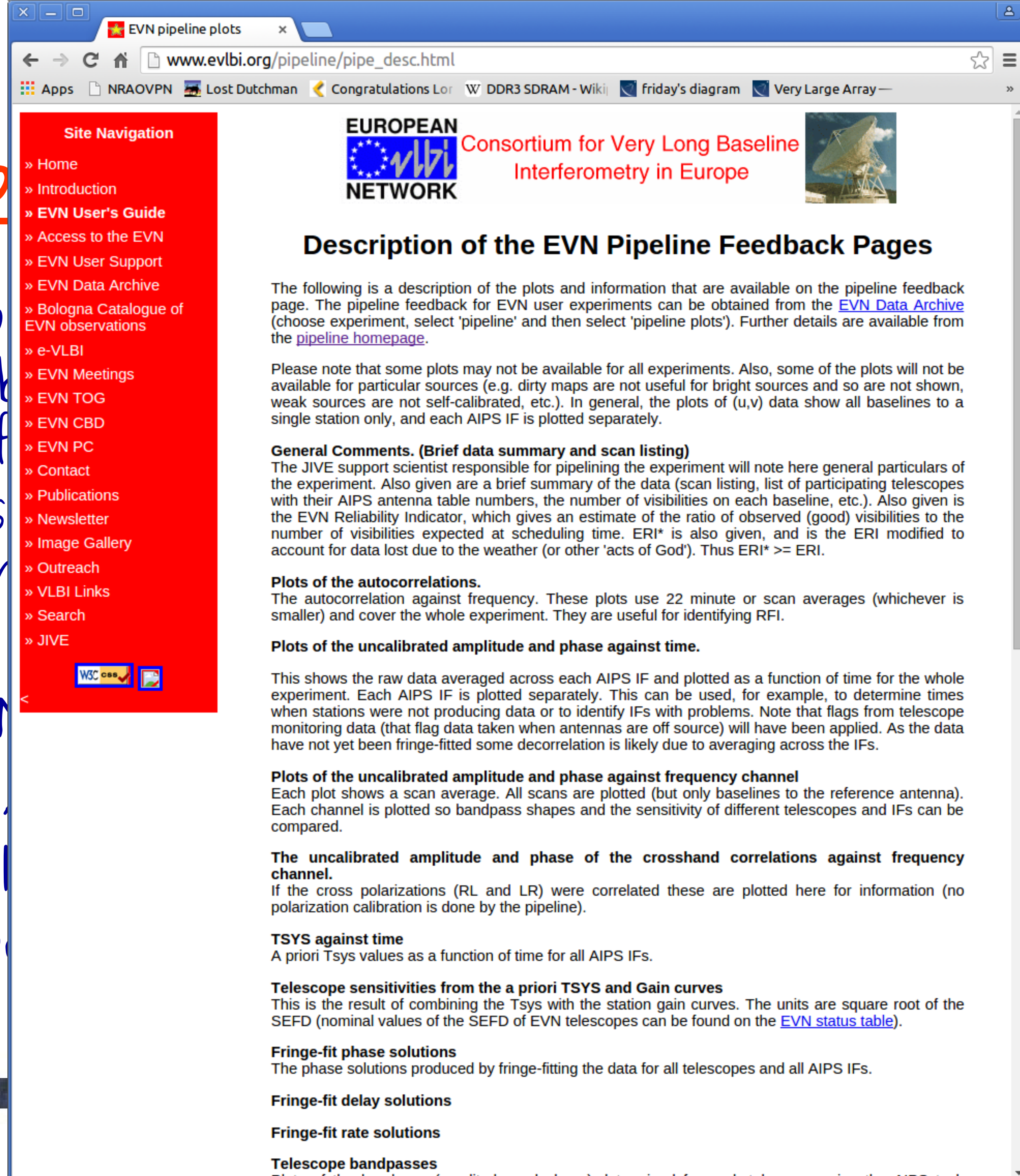
Usage

```
$ ParselTongue EVN.py template.txt
```

will run the pipeline using the given input file. A very brief guide to running the pipeline is available: pypeline_public.pdf. A description of the output of the pipeline is available at: http://www.evlbi.org/pipeline/pipe_desc.html. Any comments/questions on the pipeline should be directed to Stephen Bourke (bourke@jive.nl)

Pip

- Observato
  need to b
  variety of
  strategies
  calibratio
- Examples:
  - EVN
  - VL
  - AL
- Code, do



EVN pipeline plots

www.evlbi.org/pipeline/pipe_desc.html

Apps | NRAOVPN | Lost Dutchman | Congratulations Lo | DDR3 SDRAM - Wiki | friday's diagram | Very Large Array —

Site Navigation

- » Home
- » Introduction
- » **EVN User's Guide**
- » Access to the EVN
- » EVN User Support
- » EVN Data Archive
- » Bologna Catalogue of EVN observations
- » e-VLBI
- » EVN Meetings
- » EVN TOG
- » EVN CBD
- » EVN PC
- » Contact
- » Publications
- » Newsletter
- » Image Gallery
- » Outreach
- » VLBI Links
- » Search
- » JIVE

EUROPEAN VLBI NETWORK — Consortium for Very Long Baseline Interferometry in Europe

Description of the EVN Pipeline Feedback Pages

The following is a description of the plots and information that are available on the pipeline feedback page. The pipeline feedback for EVN user experiments can be obtained from the EVN Data Archive (choose experiment, select 'pipeline' and then select 'pipeline plots'). Further details are available from the pipeline homepage.

Please note that some plots may not be available for all experiments. Also, some of the plots will not be available for particular sources (e.g. dirty maps are not useful for bright sources and so are not shown, weak sources are not self-calibrated, etc.). In general, the plots of (u,v) data show all baselines to a single station only, and each AIPS IF is plotted separately.

General Comments. (Brief data summary and scan listing)
The JIVE support scientist responsible for pipelining the experiment will note here general particulars of the experiment. Also given are a brief summary of the data (scan listing, list of participating telescopes with their AIPS antenna table numbers, the number of visibilities on each baseline, etc.). Also given is the EVN Reliability Indicator, which gives an estimate of the ratio of observed (good) visibilities to the number of visibilities expected at scheduling time. ERI* is also given, and is the ERI modified to account for data lost due to the weather (or other 'acts of God'). Thus ERI* >= ERI.

Plots of the autocorrelations.
The autocorrelation against frequency. These plots use 22 minute or scan averages (whichever is smaller) and cover the whole experiment. They are useful for identifying RFI.

Plots of the uncalibrated amplitude and phase against time.

This shows the raw data averaged across each AIPS IF and plotted as a function of time for the whole experiment. Each AIPS IF is plotted separately. This can be used, for example, to determine times when stations were not producing data or to identify IFs with problems. Note that flags from telescope monitoring data (that flag data taken when antennas are off source) will have been applied. As the data have not yet been fringe-fitted some decorrelation is likely due to averaging across the IFs.

Plots of the uncalibrated amplitude and phase against frequency channel
Each plot shows a scan average. All scans are plotted (but only baselines to the reference antenna). Each channel is plotted so bandpass shapes and the sensitivity of different telescopes and IFs can be compared.

The uncalibrated amplitude and phase of the crosshand correlations against frequency channel.
If the cross polarizations (RL and LR) were correlated these are plotted here for information (no polarization calibration is done by the pipeline).

TSYS against time
A priori Tsys values as a function of time for all AIPS IFs.

Telescope sensitivities from the a priori TSYS and Gain curves
This is the result of combining the Tsys with the station gain curves. The units are square root of the SEFD (nominal values of the SEFD of EVN telescopes can be found on the EVN status table).

Fringe-fit phase solutions
The phase solutions produced by fringe-fitting the data for all telescopes and all AIPS IFs.

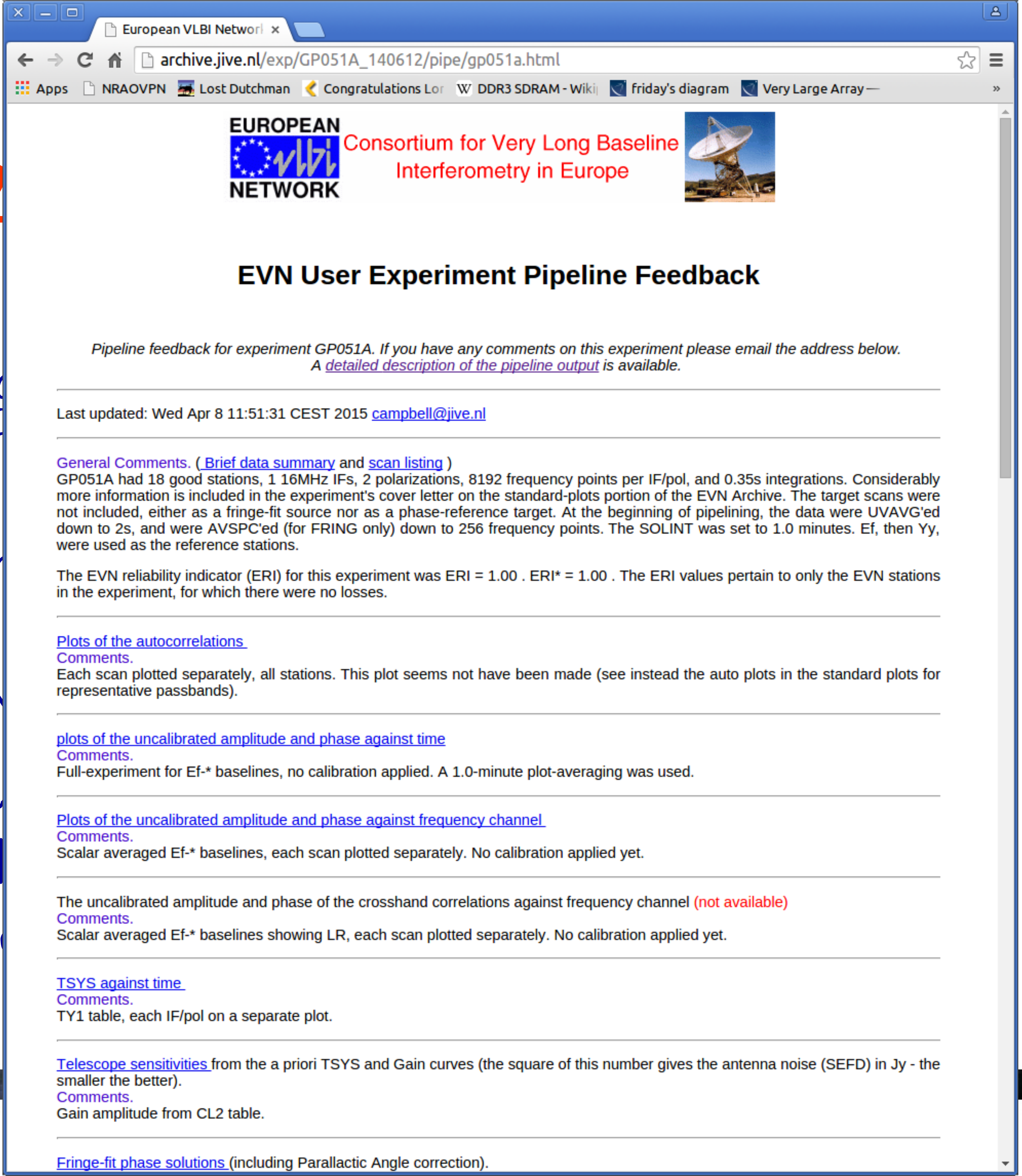
Fringe-fit delay solutions

Fringe-fit rate solutions

Telescope bandpasses
Plots of the bandpass (amplitude and phase) determined for each telescope using the AIPS task

Pip

- Observato
  need to b
  variety of
  strategies
  calibration
- Examples:
  - EVN
  - VL
  - AL
- Code, do



European VLBI Network

archive.jive.nl/exp/GP051A_140612/pipe/gp051a.html

Apps | NRAOVPN | Lost Dutchman | Congratulations Lor | DDR3 SDRAM - Wiki | friday's diagram | Very Large Array —

EUROPEAN vlbi NETWORK

Consortium for Very Long Baseline Interferometry in Europe

EVN User Experiment Pipeline Feedback

Pipeline feedback for experiment GP051A. If you have any comments on this experiment please email the address below. A detailed description of the pipeline output is available.

Last updated: Wed Apr 8 11:51:31 CEST 2015 campbell@jive.nl

General Comments. (Brief data summary and scan listing)
GP051A had 18 good stations, 1 16MHz IFs, 2 polarizations, 8192 frequency points per IF/pol, and 0.35s integrations. Considerably more information is included in the experiment's cover letter on the standard-plots portion of the EVN Archive. The target scans were not included, either as a fringe-fit source nor as a phase-reference target. At the beginning of pipelining, the data were UVAVG'ed down to 2s, and were AVSPC'ed (for FRING only) down to 256 frequency points. The SOLINT was set to 1.0 minutes. Ef, then Yy, were used as the reference stations.

The EVN reliability indicator (ERI) for this experiment was ERI = 1.00 . ERI* = 1.00 . The ERI values pertain to only the EVN stations in the experiment, for which there were no losses.

Plots of the autocorrelations
Comments.
Each scan plotted separately, all stations. This plot seems not have been made (see instead the auto plots in the standard plots for representative passbands).

plots of the uncalibrated amplitude and phase against time
Comments.
Full-experiment for Ef-* baselines, no calibration applied. A 1.0-minute plot-averaging was used.

Plots of the uncalibrated amplitude and phase against frequency channel
Comments.
Scalar averaged Ef-* baselines, each scan plotted separately. No calibration applied yet.

The uncalibrated amplitude and phase of the crosshand correlations against frequency channel (not available)
Comments.
Scalar averaged Ef-* baselines showing LR, each scan plotted separately. No calibration applied yet.

TSYS against time
Comments.
TY1 table, each IF/pol on a separate plot.

Telescope sensitivities from the a priori TSYS and Gain curves (the square of this number gives the antenna noise (SEFD) in Jy - the smaller the better).
Comments.
Gain amplitude from CL2 table.

Fringe-fit phase solutions (including Parallactic Angle correction).

NRAO

Pipelines in the field

- Observatory based pipelines for user data need to be more general due to the variety of observing projects, modes and strategies, and typically attempt flagging, calibration and "reference images"
- Examples:
  - EVN Calibration pipeline
  - VLA CASA scripted pipeline
  - ALMA pipeline
- Code, documentation "available" (support?)

NRAO

Pip

- Observato
  need to b
  variety of
  strategies
  calibration
- Examples:
  - EV
  - VL
  - AL
- Code, do

VLA Scripted Calibratio ×

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

Apps | NRAOVPN | Lost Dutchman | Congratulations Lor | DDR3 SDRAM - Wiki | friday's diagram | Very Large Array —

National Radio Astronomy Observatory
Enabling forefront research into the Universe at radio wavelengths

my.nrao.edu | Public Site | Contact Us | **Staff Login**

Search all of NRAO | Go

Home | About NRAO | Science | Facilities | Observing | Opportunities

ALMA/NAASC | VLA | GBT | VLBA | CDL

Facilities > VLA > Data Processing > Pipeline

- Proposing
- Observing
- Data Processing
- Data Archive
- Other Info for Observers
- HelpDesk
- Science

VLA Scripted Calibration Pipeline

General Description

The VLA calibration pipeline performs basic flagging and calibration using CASA. It is designed to work for Stokes I continuum data, but may work in other circumstances as well. Starting with the D-configuration, Semester 2013A, it will be run automatically at the completion of all astronomical scheduling blocks (SBs), and the resulting calibration tables and flags are saved in the archive. The pipeline products undergo quality assurance checks by NRAO staff, and investigators are notified when the calibrated data are ready for download. The calibrated visibility data are retained on disk for 15 days after the pipeline has completed, to enable investigators to download and image at their home institution. Calibrated data can also be provided after this nominal time period, by re-generating the measurement set and applying the saved calibration and flag tables.

The VLA calibration pipeline runs on each completed SB separately; there is currently no provision for it running on collections of SBs. The pipeline relies entirely on correct *scan intents* to be defined in each SB. In order for the pipeline to run successfully on an SB it must contain, *at minimum*, scans with the following intents:

1. a flux density calibrator scan that observes one of the primary calibrators (this will also be used as the delay and bandpass calibrator if no bandpass or delay calibrator is defined)
2. complex gain calibrator scans.

The SB may also contain scans to be used specifically for bandpass and delay calibration, if desired. However, if multiple fields are defined as bandpass or delay calibrators, only the first one will be used by the pipeline. Note that a single scan or field may have multiple intents specified. Scans intended to be used to set attenuators or requantizer gains should have scan intents of *setup intent*; they must not have scan intents that may result in their being used for calibration.

In overview, the VLA calibration pipeline does the following:

- Loads the data into a CASA measurement set (MS), applies Hanning smoothing to them, and obtains information about the observing set-up from the MS
- Applies online flags and other deterministic flags (shadowed data, end channels of sub-bands, etc.)
- Prepares models for primary flux density calibrators
- Derives pre-determined calibrations (antenna position corrections, gain curves, atmospheric opacity corrections, requantizer gains, etc.)
- Iteratively determines initial delay and bandpass calibrations, including flagging of RFI and some automated identification of system problems
- Derives initial gain calibration, and derives the spectral index of the bandpass calibrator
- Derives final delay, bandpass, and gain calibrations, and applies them to the data

Contents

Pip

- Observato
  need to b
  variety of
  strategies
  calibration
- Examples:
  - EVN
  - VL
  - AL
- Code, do

VLA Scripted Calibratio ×

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

Apps | NRAOVPN | Lost Dutchman | Congratulations Lor | DDR3 SDRAM - Wiki | friday's diagram | Very Large Array —

Running the Pipeline

Stokes I Continuum

To run the script on datasets that contain only continuum data (64 MHz or 128 MHz spectral windows):

1. Put your data (SDM-BDF or measurement set) in its own directory for processing. For example:

```
mkdir myVLAdata
mv mySDM myVLAdata/
cd myVLAdata
```

2. Start casapy from myVLAdata (this is important — do not try to run the pipeline from a different directory by giving it the full path to a dataset, as some of the CASA tasks require the MS to be co-located with its associated gain tables; also, do not try to run the pipeline from inside the SDM-BDF or MS directories themselves). It is also important that a fresh instance of CASA is started from the directory that will contain the SDM-BDF and MS, rather than using an existing instance of CASA and using "cd" to move to a new directory from within CASA, as the output plots will then end up in the wrong place and potentially overwrite your previous pipeline results.
3. From the CASA prompt, type:

```
execfile('/home/mymachine/pipe_scripts/EVLA_pipeline.py')
```

4. The pipeline will then prompt you for the SDM-BDF name; if you have only an MS, give it the root name of the MS (i.e., omit any '.ms').
5. The pipeline will then prompt you for whether or not you want to use Hanning smoothing (this can be important for strong, narrow-band RFI, but there are some situations where it is not desirable: for low frequencies in A-configuration it may increase bandwidth smearing, and for spectral line observations it will make the spectral resolution worse).
6. Go and make some coffee.
7. The pipeline will automatically generate a QA2 score of Pass, Partial, or Fail for the pipeline as a whole and for each step along the way. For some basic help interpreting your QA2 score, please see our QA2 interpretation page.

Spectral Line Data

If your calibrators are strong enough that the heuristics in the VLA calibration pipeline will work on narrower bandwidths then some simple edits to the master EVLA_pipeline.py script is all that is needed in order for it to work on a spectral line datsaset. In particular, you will want to comment out the call to flagdata() that is executing the flagging task with mode='rflag' in the EVLA_pipe_targetflag.py script. This runs the target through an algorithm that searches for and flags RFI (rflag), and may therefore remove your spectral line as well. You may also want to answer "n" to the Hanning smoothing option, and depending on the strength of your line, you may want to modify the inputs to "statwt" to exclude channels containing line emission.

Mixed set-ups

In the case where a mixed continuum/spectral line set-up has been used, or multiple receiver bands have been observed, it may be the case that a single pipeline heuristic (e.g., gain calibration solution interval) is not appropriate for the entire dataset. In this case, the MS can be split by correlator set-up/receiver band (typically specified by selecting on spectral windows or scans) after applying the online flags, and the pipeline run on the split datasets individually. To do this:

1. Copy a version of EVLA_pipeline.py into your local directory being used for data reduction
2. Edit EVLA_pipeline.py, commenting out all "execfile" calls *after* EVLA_pipe_flagall.py
3. Run the pipeline on the full SDM-BDF/MS through EVLA_pipe_flagall.py, to apply all online flags. Include Hanning smooth at this point, if you are going to use it: execfile('EVLA_pipeline.py')
4. Using the <SDMname>.listobs output, identify the groups of spws and/or scans to split (e.g., all spws associated with a particular observing band, or all spws with a particular spectral set-up)

NRAO

Pip

- Observato
need to b
variety of
strategies
calibration
- Examples:
  - EVN
  - VL
  - AL
- Code, do

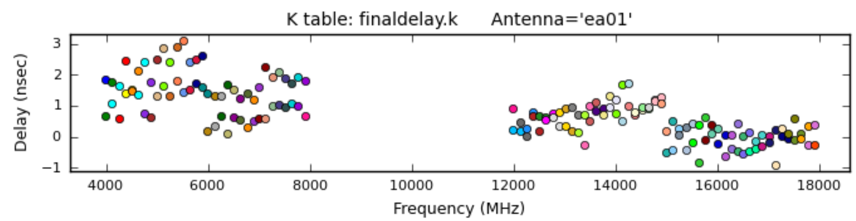
VLA Pipeline QA2

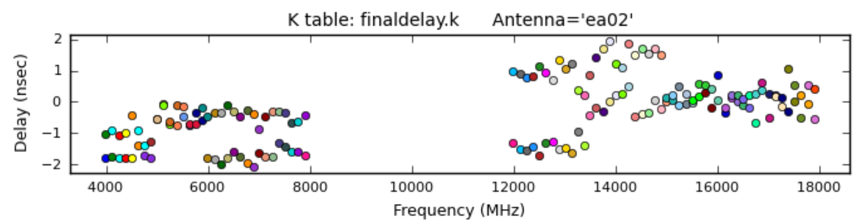
www.aoc.nrao.edu/~lsjouwer/weblog049/qa2.html

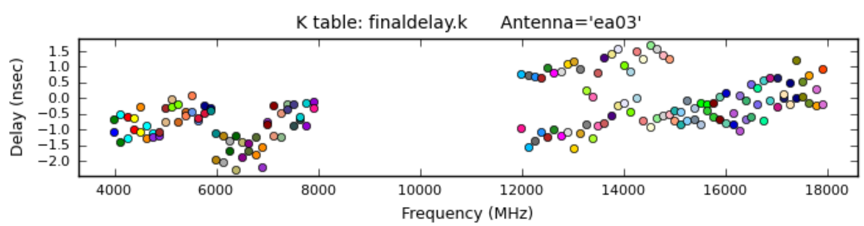
Apps | NRAOVPN | Lost Dutchman | Congratulations Lor | DDR3 SDRAM - Wiki | friday's diagram | Very Large Array —

Splash Page	Observation Summary	QA2	Task Summary

Fraction of zeroes: 0.00107054166039
Total fraction of on-source data flagged before calibration (NB: expect approx. 0.1 from flagging of end channels): 0.104364320562
Final fraction of on-source data flagged: 0.221832400954
QA2 scores for each step of pipeline: link
Overall QA2 score: Pass
There are no missing scans
Gain solution interval used for initial delay calibration: int
Gain solution interval used for BP calibrator: int
Short/long gain solution interval used for gain calibrator: 3.0s/204.020001342s
Fitted flux densities: link
Final delays [abs(max.delay) = 11.9774646759 ns]:



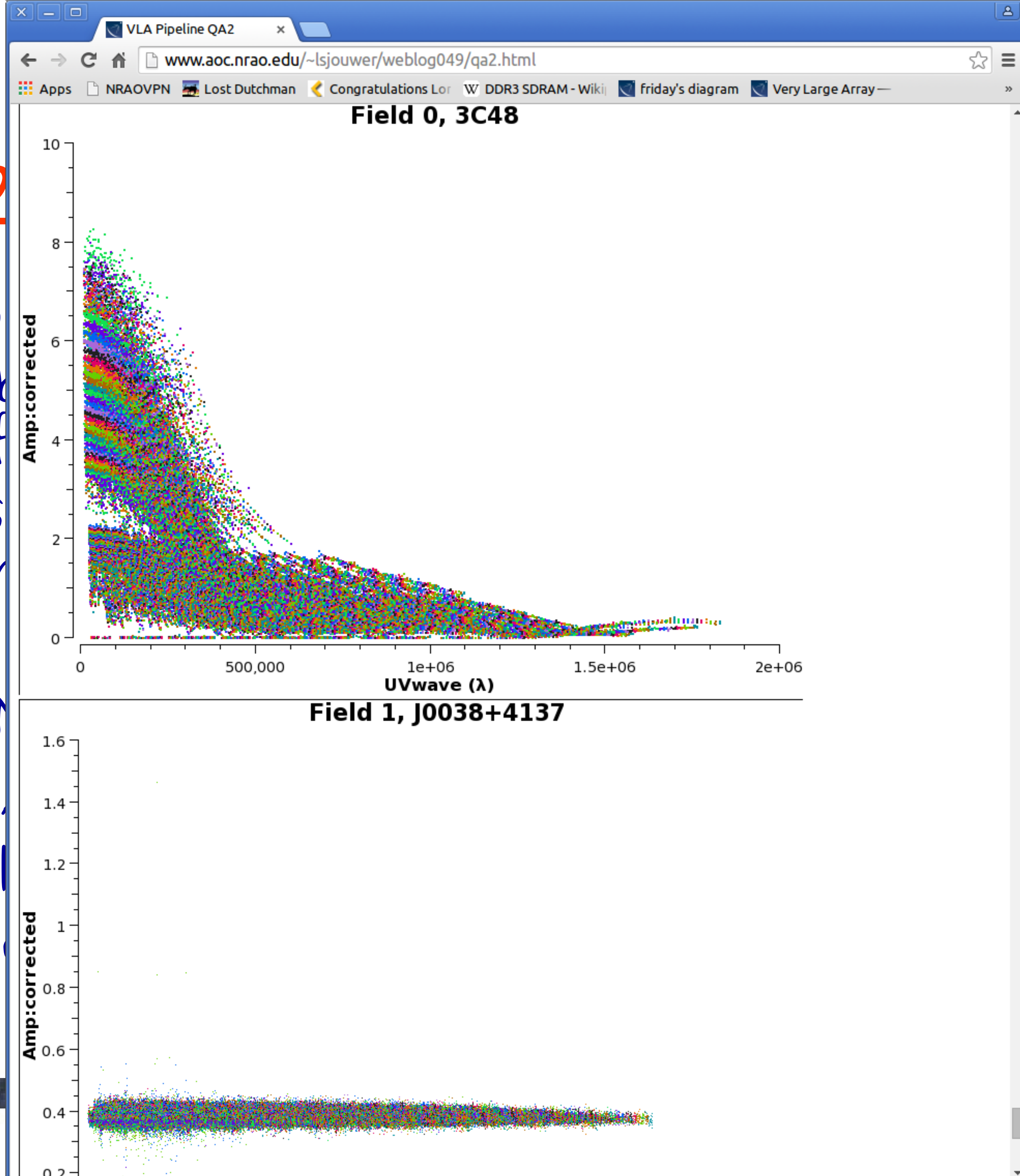




NRAO

Pip

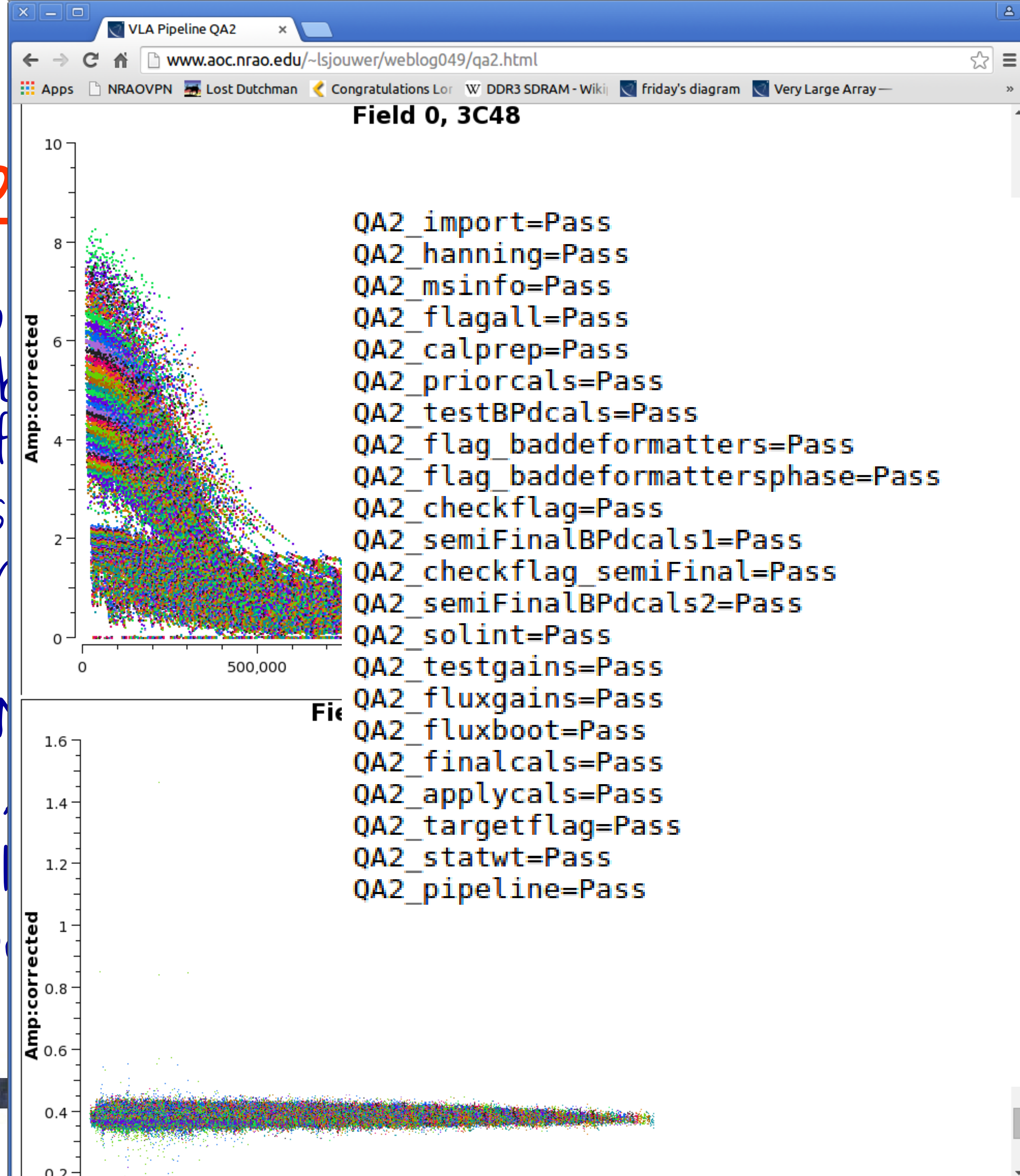
- Observato
  need to b
  variety of
  strategies
  calibration
- Examples:
  - EVN
  - VL,
  - ALI
- Code, do



NRAO

Pip

- Observato
  need to b
  variety of
  strategies
  calibratio
- Examples:
  - EVI
  - VL
  - ALI
- Code, do



NRAO

Pipelines in the field

- Observatory based pipelines for user data need to be more general due to the variety of observing projects, modes and strategies, and typically attempt flagging, calibration and "reference images"
- Examples:
  - EVN Calibration pipeline
  - VLA CASA scripted pipeline
  - ALMA pipeline
- Code, documentation "available" (support?)

NRAO

Pipelines in the field

- Observatory based pipelines for user data need to be more general due to the variety of observing projects, modes and strategies, and typically attempt flagging, calibration and "reference images"
- Examples:
  - EVN Calibration pipeline
  - VLA CASA scripted pipeline
  - ALMA pipeline – **Liz in a few…**
- Code, documentation "available" (support?)

NRAO

Pipelines in the field

- Pipelines delivered as part of a data reduction package, like CASA and AIPS for a specific array or general use
- General examples in AIPS and CASA:
  - VLBAPIPE/VLBARUN in AIPS
  - VLARUN/DOOSRO/PIPEAIPS in AIPS
  - VLA pipeline in CASA
  - ALMA pipeline in CASA
- Code and documentation distributed with the package, and support is available!

NRAO

Pipelines i

- Pipelines delivered a reduction package, l for a specific array
- General examples in
  - VLBAPIPE/VL
  - VLARUN/DOOS
  - VLA pipeline i
  - ALMA pipeline
- Code and documenta the package, and su

```
File  Edit  View  Search  Terminal  Help
>
>inp vlbarun
AIPS 3: VLBARUN: Applies amplitude and phase calibration to VLBA data !
AIPS 3: Adverbs     Values                       Comments
AIPS 3: ----------------------------------------------------------------
AIPS 3:
AIPS 3:          To run this procedure you MUST type RUN VLBAUTIL and
AIPS 3:          RUN VLBARUN first to define the procedures in AIPS.
AIPS 3:
AIPS 3:                                          -- Load the data from disk --
AIPS 3: DATAIN     *all ' '                      Disk file name
AIPS 3: DOUVCOMP     -1                          Compress data? Use -1, 0 or 1
AIPS 3: OUTNAME    ' '                           File name (name)
AIPS 3: OUTDISK       1                          Working disk with ample space
AIPS 3:                                          -- OR existing file --
AIPS 3: INNAME     ' '                           Input file name
AIPS 3: INCLASS    ' '                           Input file class
AIPS 3: INSEQ         0                          Input file sequence number
AIPS 3: INDISK        0                          Disk number for input file
AIPS 3:                                          -----------------------------
AIPS 3: OPTYPE     ' '                           For CONT, PSEU or SPEC (LINE)
AIPS 3:                                          -----------------------------
AIPS 3: CLINT         0                          CL table interval in minutes.
AIPS 3: CHREFANT   ' '                           Reference antenna NAME
AIPS 3: TIMERANG   *all 0                        A good bandpass scan, or use
AIPS 3:                                            scan# in TIMERANG(1)+rest=0
AIPS 3: INVER         0                          PC table to use
AIPS 3:                                          -1 => don't use Pulse cals
AIPS 3:                                              do manual phase cal
AIPS 3: CALSOUR    *all ' '                      List ALL Fring fit & CAL sour
AIPS 3:                                            FIRST is bandpass calib
AIPS 3: SOURCES    *all ' '                      ONLY FOR PHASE REFERENCING:
AIPS 3:                                            Source *PAIRS* to calibrate
AIPS 3:                                            START each pair with phase
AIPS 3:                                            referencing calibrator, if
AIPS 3:                                            2nd='*': all non-CALSOUR
AIPS 3:                                            are phase-referenced to 1st
AIPS 3:                                          -----------------------------
AIPS 3: SOLINT        0                          Time interval for fringe-fit
AIPS 3:                                          -----------------------------
AIPS 3: IMSIZE        0           0              Image size for target SOURCES
AIPS 3:                                            =-1 for no images
AIPS 3: FACTOR        0                          CALSOUR IMSIZE (FACTORx128)
AIPS 3:                                          -----------------------------
AIPS 3:          Hint: when setting directory and e-mail lower-case
AIPS 3:          letters can be retained by NOT using the close quote.
AIPS 3: DOPLOT        0                          <= 0 no plots;=1 some plots
AIPS 3:                                          >1 huge number of plots
AIPS 3: OUTFILE    *all ' '                      DIRECTORY FOR HTML AND PLOT
AIPS 3:                                            FILES: if you want an html
AIPS 3:                                            file with plots please set
AIPS 3:                                            the dir where the html and
AIPS 3:                                            plot files can be put.
AIPS 3: OUTTEXT    *all ' '                      E-MAIL ADDRESS: if you want
AIPS 3:                                            an e-mail when the script
AIPS 3:                                            is complete, set the
AIPS 3:                                            address here.
AIPS 3: BADDISK    *all 0                        Disks to avoid for scratch
>
>
>
>
>
>
>
>
```

NRAO

File Edit View Search Terminal Help

```
 euro      AIPS (31DEC15)     12     04-SEP-2015  09:12:22     Page     1
Explanation of VLBARUN in AIPS version 31DEC15
VLBARUN
Type: Procedure
Use:  VLBARUN is the procedure that uses the VLBA calibration procedures
      (VLBAUTIL) to calibrate VLBA data.  See the explain file for a detailed
      description of the procedure.
      Simple visibility flagging, bandpass calibration, and
      ionospheric (total electron content) corrections are applied before
      fringe fitting and averaging.  It may attempt to self-cal the
      calibrators and will image the targets.

NOTE: currently this procedure is intended for simple experiments!

$---------------------------------------------------------------------------

      Type RUN VLBAUTIL and RUN VLBARUN to define the VLBARUN procedures.

      The procedure is run by typing : (GO) VLBARUN
      The procedure will check the inputs (as far as it can) to avoid later
      interruptions, and then proceed directly, if the data is already on
      disk.

CLEAN STARTING CONDITIONS
      It is best to use one project per user number, as otherwise you may get
      into trouble if different projects use the same source names.
      It is not recommended to run two pipelines with the same user number
      simultaneously, nor overlapping at any time during the processes.

      Clean up all files resulting from previous attempts of running the
      procedure, except for the original UVDATA and perhaps, when restarting
      a specific frequency-ID, also not the F(X)POL files. When restarting
      from disk, make sure the UV data has no extra SN, CL, FG etc tables.
      Use the procedure P_RESTART (which is defined in VLBARUN) to do this.

      If you  wish to flag some additional data (because the procedure did
      not flag all the bad data automatically), make sure you put the flags
      in FG table number 1.

      If you have any antennas that do not have Tsys or Gain curves missing
      from the data, please load them with ANTAB into TY and GC tables #1.

      It is a VERY good idea to run TASAV before you do this to keep the
      original TY, GC, BL and also FG tables number 1 in case of disasters.

      Make sure there is enough disk space available on OUTDISK, about 4 to
      5 times the expected compressed UVDATA set. For spectral line probably
      you need much more because it will make spectral line image cubes.


  DATAIN......48-character name of the disk file from which to read a
              FITS file.  It must be in the form
                  <logical>:<file name>
              or
                  <node>::<logical>:<file name>
              where <node> is the remote computer name, <logical> is the
              environment variable (logical name) for the disk area in
              which the file named <file name> is stored.  <node> is
              usually omitted when the file is local to the current
              computer.  If DATAIN is not found, the task will try
              DATAIN with the character 1 appended.
              Beginning 2003-Oct-16 FITLD can read more than one disk
              file at a time.  In that case, they must all have the
              same name except that the last letter(s) are the
AIPS 3: ** press RETURN for more, enter Q or next line to quit print **
#
```

Edit View Search Terminal Help

```
> vlbarun
; 3: VLBARUN: Applies amplitude and phase calibration to VLBA data !
; 3: Adverbs     Values                       Comments
; 3: ----------------------------------------------------------------
; 3:
; 3:          To run this procedure you MUST type RUN VLBAUTIL and
; 3:          RUN VLBARUN first to define the procedures in AIPS.
; 3:
; 3:                                         -- Load the data from disk --
; 3: DATAIN     *all ' '                     Disk file name
; 3: DOUVCOMP     -1                         Compress data? Use -1, 0 or 1
; 3: OUTNAME    ' '                          File name (name)
; 3: OUTDISK       1                         Working disk with ample space
; 3:                                         -- OR existing file --
; 3: INNAME     ' '                          Input file name
; 3: INCLASS    ' '                          Input file class
; 3: INSEQ         0                         Input file sequence number
; 3: INDISK        0                         Disk number for input file
; 3:                                         ----------------------------
; 3: OPTYPE     ' '                          For CONT, PSEU or SPEC (LINE)
; 3:                                         ----------------------------
; 3: CLINT         0                         CL table interval in minutes.
; 3: CHREFANT   ' '                          Reference antenna NAME
; 3: TIMERANG   *all 0                       A good bandpass scan, or use
; 3:                                           scan# in TIMERANG(1)+rest=0
; 3: INVER         0                         PC table to use
; 3:                                         -1 => don't use Pulse cals
; 3:                                             do manual phase cal
; 3: CALSOUR    *all ' '                     List ALL Fring fit & CAL sour
; 3:                                             FIRST is bandpass calib
; 3: SOURCES    *all ' '                     ONLY FOR PHASE REFERENCING:
; 3:                                           Source *PAIRS* to calibrate
; 3:                                           START each pair with phase
; 3:                                           referencing calibrator, if
; 3:                                           2nd='*': all non-CALSOUR
; 3:                                           are phase-referenced to 1st
; 3:                                         ----------------------------
; 3: SOLINT        0                         Time interval for fringe-fit
; 3:                                         ----------------------------
; 3: IMSIZE        0           0             Image size for target SOURCES
; 3:                                           =-1 for no images
; 3: FACTOR        0                         CALSOUR IMSIZE (FACTORx128)
; 3:                                         ----------------------------
; 3:          Hint: when setting directory and e-mail lower-case
; 3:          letters can be retained by NOT using the close quote.
; 3: DOPLOT        0                         <= 0 no plots;=1 some plots
; 3:                                         >1 huge number of plots
; 3: OUTFILE    *all ' '                     DIRECTORY FOR HTML AND PLOT
; 3:                                           FILES: if you want an html
; 3:                                           file with plots please set
; 3:                                           the dir where the html and
; 3:                                           plot files can be put.
; 3: OUTTEXT    *all ' '                     E-MAIL ADDRESS: if you want
; 3:                                           an e-mail when the script
; 3:                                           is complete, set the
; 3:                                           address here.
; 3: BADDISK    *all 0                       Disks to avoid for scratch
```

Pipelines in the field

- Pipelines delivered as part of a data reduction package, like CASA and AIPS for a specific array or general use
- General examples in AIPS and CASA:
  - VLBAPIPE/VLBARUN in AIPS
  - VLARUN/DOOSRO/PIPEAIPS in AIPS
  - VLA pipeline in CASA
  - ALMA pipeline in CASA
- Code and documentation distributed with the package, and support is available!

NRAO

Pipelines i

- Pipelines delivered a reduction package, l for a specific array
- General examples in
  - VLBAPIPE/VL
  - VLARUN/DOOS
  - VLA pipeline i
  - ALMA pipeline
- Code and documenta the package, and su

```
File  Edit  View  Search  Terminal  Help
>inp pipeaips
AIPS 3: PIPEAIPS: Amp+phase calibration for conn. elem. interferometers
AIPS 3: Adverbs     Values                     Comments
AIPS 3: ----------------------------------------------------------------
AIPS 3:
AIPS 3:          ** Type RUN PIPEAIPS to load this procedure **
AIPS 3:
AIPS 3: WORKDISK      0                        Working disk, thus in/outdisk
AIPS 3: CATNUM        0                        Catalog number of the UV-file
AIPS 3: INNAME       ' '                       Input UV file name (CATNUM<1)
AIPS 3: INCLASS      ' '                       Input UV file class(CATNUM<1)
AIPS 3: INSEQ         0                        Input UV file seq. (CATNUM<1)
AIPS 3: TINT          0                        Re-average visibilities (sec)
AIPS 3:                                         or corr.avg. time if known
AIPS 3: FASTSW        0                        > 0 Correct fast-sw source
AIPS 3:                                            names
AIPS 3: VLANTCOR      0                        Run VLANT? Only do once!
AIPS 3: AUTOFLAG      0                        Level of automatic flags used
AIPS 3:                                        < 0: no automatic flagging.
AIPS 3:                                        = 0: default FLAGR
AIPS 3:                                        = 1: flag beginning of scans
AIPS 3: PHAINT        0                        phase solution interval (min)
AIPS 3: AMPINT        0                        ampl. solution interval (min)
AIPS 3: BASEBAND      0                        Data has N equal basebands:
AIPS 3:                                         see help and FRING aparm(5)
AIPS 3: BPNORM        0                        Bandpass 'divide-by' control
AIPS 3:                                         see help, BPASS bpassprm(5)
AIPS 3: REFANT        0                        Reference antenna number
AIPS 3: DOMODEL      -1                        > 0 Use standard flux
AIPS 3:                                            calibrator models
AIPS 3:                                        Note that most standard flux
AIPS 3:                                        calibrators have models but
AIPS 3:                                        the absence of a model will
AIPS 3:                                        make this CRASH. (see HELP)
AIPS 3: UVRANGE       0           0            UV range for flux calibrator
AIPS 3:                                        (may be used if no model)
AIPS 3:          ** use next 2 lines if flux calib. is NOT standard **
AIPS 3: AMPCAL       ' '                       Alternative flux calib. name
AIPS 3: FLUX          0                        Flux calib. total flux dens.
AIPS 3: PHACAL      *all ' '                   Phase calibrators
AIPS 3:                                        '*' = any CALCODE (continuum)
AIPS 3:                                         All others are your targets
AIPS 3: BNDCAL      *all ' '                   Bandpass calibrators (max 5)
AIPS 3: NOPAUSE       0                        > 0 no pause after GETJY
AIPS 3: AUTOPLOT      0                        > 0 make diagnostic plots
AIPS 3:          ** the following lines are for auto-imaging **
AIPS 3: DOIMAGES      0                        > 0 apply calibration and
AIPS 3:                                            image
AIPS 3: IMGTYPE      ' '                       'CONT', 'PSEU', or 'LINE' for
AIPS 3:                                         continuum (single image),
AIPS 3:                                         pseudo cont (image per IF)
AIPS 3:                                         or line (full image cube)
AIPS 3: ARRYSIZE      0                        Max baseline in kilometers
AIPS 3:                                        = 0 let procedure find array
AIPS 3: IMSIZE        0           0            Square size of image (pixels)
AIPS 3: NITER         0                        Max number of iterations/chan
AIPS 3:                                        = 0 recommended for LINE
AIPS 3: CUTOFF        0                        Clean threshold (Jy)
AIPS 3: ALLIMG        0                        > 0 also image continuum
AIPS 3:                                            calibrators
AIPS 3:                                        > 1 line calibrators too
AIPS 3: DOEVAUV       0                        Only for IMGTYPE='PSEU' and
AIPS 3:                                        if > 0 then run 'EVAUV' too
AIPS 3:          ** interactive self calibration mode
AIPS 3: ** press RETURN for more, enter Q or next line to quit print **
#
```

File Edit View Search Terminal Help

```
 euro      AIPS (31DEC15)     12     04-SEP-2015  09:13:10    Page    1
Explanation of PIPEAIPS in AIPS version 31DEC15
PIPEAIPS
Type: Procedure
Use:  PIPEAIPS is a procedure that does quick and dirty VLA
      calibration and imaging of line and continuum data,
      including high frequency data.  Full polarization
      calibration is not included.

NOTE: This PIPEAIPS code originates from earlier VLARUN/DOOSRO versions

      To load the procedure into AIPS type:
             (this is only required once per AIPS session)
         > RESTORE 0 (recommended, not required)
         > RUN PIPEAIPS
         > COMPRESS
      To review inputs type:
         > TASK'PIPEAIPS';INPUTS
      To execute it type:
         > PIPEAIPS

Adverbs:
CATNUM, or all of INNAME, INCLASS, INSEQ (and WORKDISK for both):
      The catalog number of the file to calibrate; for VLA data loaded
      with BDF2AIPS this would be UVEVLA for a yet unprocessed file.
      Add flagging info on missing receivers, etc., already here in FG1.
      PIPEAIPS converts this into a SPLAT0/SPLATL pair to keep the data
      in spectral window form but also using a channel 0 for speed.  If a
      a first instance of PIPEAIPS has run, the 'SPLAT0' should be chosen.
      If INNAME etc are used, then set CATNUM to a non-positive value (<=0)
      i.e. CATNUM=0;GETNAME <catno> Do not use both CATNUM and IN<*>

TINT:
      Some (VLA) data sets have unnecessary short (i.e., 1 sec) visibility
      integrations.  Set this to a larger value (>=1, typically an integer
      multiple of the original correlator integration interval) to decrease
      the data size by this factor.  Leave zero to do no averaging, though
      if the correlator averaging time is known, setting it to that interval
      (in sec) will take out the guess work in finding a value from the data.

FASTSW:
      Sometimes, with fast-switching, single calibrator may have more than
      one name.  Set FASTSW positive to check the positions (closer than
      3 mas in R.A. and DEC.), qualifier and calcode.  If sources have
      the same position, PIPEAIPS will rename them all with the shortest
      name.  This will prevent multiple images of the same source.  Up to
      100 source names can be modified.

VLANTCOR:
      Skip running VLANT if < 0 or for an input file with class 'UVLANT' or
      for an input file with class 'SPLAT*' (i.e. 'SPLAT0' and 'SPLATL').
      VLANT should be run only once so set negative if you know it was run.
      If set to 0, the pipeline runs VLANT and will change the class of the
      input file (e.g. 'UVEVLA' or 'UVDATA') to 'UVLANT' and thus skip it on
      successive runs of PIPEAIPS.

AUTOFLAG:
      < 0 -> no automated flagging.
      NOTE: if an FG table is detected, it WILL NOT DO any form of automated
      flagging on the data.  You may not want to do autoflagging when you
      have very short scans (i.e.  in fast-switching mode) and flag by hand.
      The highest FG-table is always kept.
      0 -> Perform FLAGR (OPTYPE'TIME') on the raw multi-source data set
          PIPEAIPS estimates the integration time and SOLINT, in FLAGR, is set
AIPS 3: ** press RETURN for more, enter Q or next line to quit print **
#
```

Edit View Search Terminal Help

```
> pipeaips
; 3: PIPEAIPS: Amp+phase calibration for conn. elem. interferometers
; 3: Adverbs     Values                  Comments
; 3: ----------------------------------------------------------------
; 3:
; 3:          ** Type RUN PIPEAIPS to load this procedure **
; 3:
; 3: WORKDISK      0                     Working disk, thus in/outdisk
; 3: CATNUM        0                     Catalog number of the UV-file
; 3: INNAME      ' '                     Input UV file name (CATNUM<1)
; 3: INCLASS     ' '                     Input UV file class(CATNUM<1)
; 3: INSEQ         0                     Input UV file seq. (CATNUM<1)
; 3: TINT          0                     Re-average visibilities (sec)
; 3:                                      or corr.avg. time if known
; 3: FASTSW        0                     > 0 Correct fast-sw source
; 3:                                         names
; 3: VLANTCOR      0                     Run VLANT? Only do once!
; 3: AUTOFLAG      0                     Level of automatic flags used
; 3:                                     < 0: no automatic flagging.
; 3:                                     = 0: default FLAGR
; 3:                                     = 1: flag beginning of scans
; 3: PHAINT        0                     phase solution interval (min)
; 3: AMPINT        0                     ampl. solution interval (min)
; 3: BASEBAND      0                     Data has N equal basebands:
; 3:                                      see help and FRING aparm(5)
; 3: BPNORM        0                     Bandpass 'divide-by' control
; 3:                                      see help, BPASS bpassprm(5)
; 3: REFANT        0                     Reference antenna number
; 3: DOMODEL      -1                     > 0 Use standard flux
; 3:                                         calibrator models
; 3:                                     Note that most standard flux
; 3:                                     calibrators have models but
; 3:                                     the absence of a model will
; 3:                                     make this CRASH. (see HELP)
; 3: UVRANGE       0            0        UV range for flux calibrator
; 3:                                     (may be used if no model)
; 3:          ** use next 2 lines if flux calib. is NOT standard **
; 3: AMPCAL      ' '                     Alternative flux calib. name
; 3: FLUX          0                     Flux calib. total flux dens.
; 3: PHACAL     *all ' '                 Phase calibrators
; 3:                                     '*' = any CALCODE (continuum)
; 3:                                      All others are your targets
; 3: BNDCAL     *all ' '                 Bandpass calibrators (max 5)
; 3: NOPAUSE       0                     > 0 no pause after GETJY
; 3: AUTOPLOT      0                     > 0 make diagnostic plots
; 3:          ** the following lines are for auto-imaging **
; 3: DOIMAGES      0                     > 0 apply calibration and
; 3:                                         image
; 3: IMGTYPE     ' '                     'CONT', 'PSEU', or 'LINE' for
; 3:                                      continuum (single image),
; 3:                                      pseudo cont (image per IF)
; 3:                                      or line (full image cube)
; 3: ARRYSIZE      0                     Max baseline in kilometers
; 3:                                     = 0 let procedure find array
; 3: IMSIZE        0            0        Square size of image (pixels)
; 3: NITER         0                     Max number of iterations/chan
; 3:                                     = 0 recommended for LINE
; 3: CUTOFF        0                     Clean threshold (Jy)
; 3: ALLIMG        0                     > 0 also image continuum
; 3:                                         calibrators
; 3:                                     > 1 line calibrators too
; 3: DOEVAUV       0                     Only for IMGTYPE='PSEU' and
; 3:                                     if > 0 then run 'EVAUV' too
; 3:          ** interactive self calibration mode
; 3: ** press RETURN for more, enter Q or next line to quit print **
```

Pipelines in the field

- Pipelines delivered as part of a data reduction package, like CASA and AIPS for a specific array or general use
- General examples in AIPS and CASA:
  - VLBAPIPE/VLBARUN in AIPS
  - VLARUN/DOOSRO/PIPEAIPS in AIPS
  - VLA pipeline in CASA
  - ALMA pipeline in CASA
- Code and documentation distributed with the package, and support is available!

NRAO

Pipelines in the field

- Pipelines delivered as part of a data reduction package, like CASA and AIPS for a specific array or general use
- General examples in AIPS and CASA:
  - VLBAPIPE/VLBARUN in AIPS
  - VLARUN/DOOSRO/PIPEAIPS in AIPS
  - VLA pipeline in CASA
  - ALMA pipeline in CASA (Liz!)
- Code and documentation distributed with the package, and support is available!

NRAO

Pipelines in the field

- Private user "pipelines"
  - May be modified versions from the previously mentioned general pipelines
  - Sometimes just a "copy/paste" script
  - For specific observations/project, may use inflexible hard-coded parameters
  - Code may be useful for sharing
    - e.g. VIPS, AIPSLite
- None, including observatory and package ones, handle polarization data well (yet)

CASA pipelines

- CASA relatively new package (well…)
- Premier data reduction package for ALMA, officially also for current VLA
- Still under active development, but pipelines are being pushed as high priority for user (and operations)

- Python scriptable

CASA pipelines

- Examples:
  - VLA CASA Calibration pipeline
    In: VLA/archive data or ms
    Aim: RFI flagging, full calibration
    Out: calibrated data, QA reports
  - ALMA pipeline
    In: ALMA/archive data
    Aim: full calibration,
    "science ready / reference images"
    Out: calibrated data, "S.R." images, QA

AIPS pipelines

- Older (35yr), mature (still evolving) package
- Uses simple POPS interpreter
- Full suite of visibility and image analysis tasks supported

- Kludge for OS interactions (sh, csh)
- Hooks with Obit package (extra algorithms)
- ParselTongue! (deep-level python scripting)

NRAO

AIPS pipelines

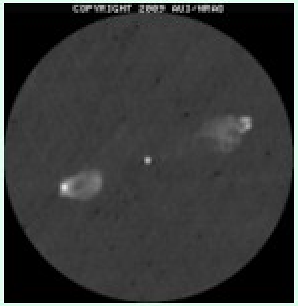
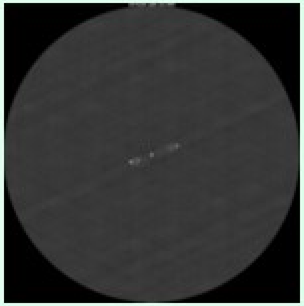
- Examples:
  - EVN Calibration pipeline
    - In: EVN/archive data
    - Aim: flagging, some self-calibration
    - Out: crude 1st order maps, QA reports
  - VLBARUN
    - In: user loaded flagged VLBI/A data
    - Aim: full calibration
    - Out: calibration to maps – user pipeline!

AIPS pipelines

- **End-to-end** example:
  - NRAO VLA Archive Survey (NVAS):
    - 30 years of VLA archive
    - mixed bag of observing runs
    - user pipeline (VLARUN)
    - single input parameter (data set)
    - from archive to indexed web pages
  - ! not infallible (simple flagging only)
    but good starting point for next step

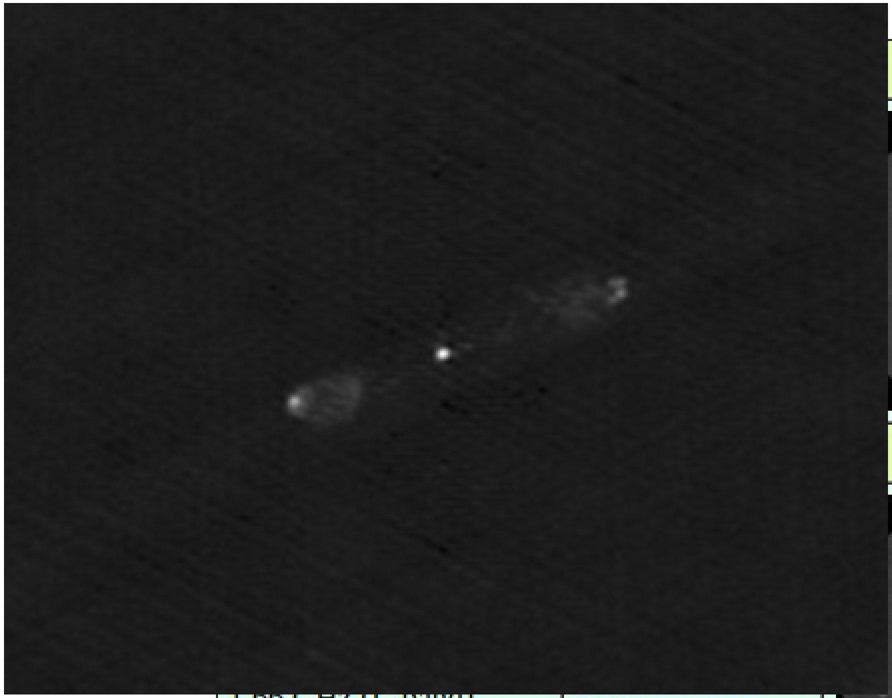
NRAO

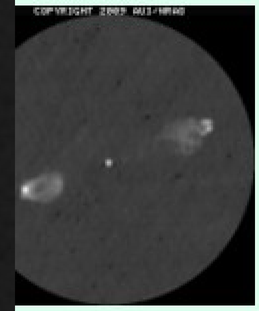
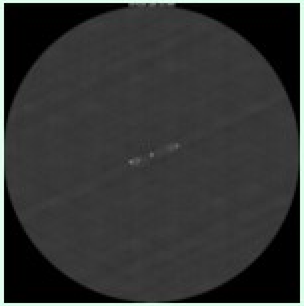
- End

Basic Info	UV Info	JPEG	Image Info	Links
15h14m 0.0s 36d50'51" (J2000 image center) 1982 MAR 4 1.49 GHz (L-band) Version: 2009 Dec 09	Full polarization A/A configuration 27 antennas subarray 1 Calibrated uv-FITS 5.8MBytes		Stokes I Beam = 1.17 arcsec fov: R = 39.0 arcsec rms = 1.26 milliJy Download FITS image 397KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
Basic Info	**UV Info**	**JPEG**	**Image Info**	**Links**
15h14m 0.0s 36d50'51" (J2000 image center) 1983 MAY 2 1.66 GHz (L-band) Version: 2009 Nov 01	Full polarization C/C configuration 27 antennas subarray 1 Calibrated uv-FITS 726KBytes		Stokes I Beam = 14.8 arcsec fov: R = 10.5 arcmin rms = 421. microJy Download FITS image 489KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
Basic Info	**UV Info**	**JPEG**	**Image Info**	**Links**
15h14m 0.0s 36d50'51" (J2000 image center) 1993 MAY 6 4.89 GHz (C-band) Version: 2007 Nov 17	Full polarization B/B configuration 27 antennas subarray 1 Calibrated uv-FITS 8.1MBytes		Stokes I Beam = 1.27 arcsec fov: R = 2.83 arcmin rms = 89.9 microJy Download FITS image 4.1MBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
Basic Info	**UV Info**	**JPEG**	**Image Info**	**Links**

!

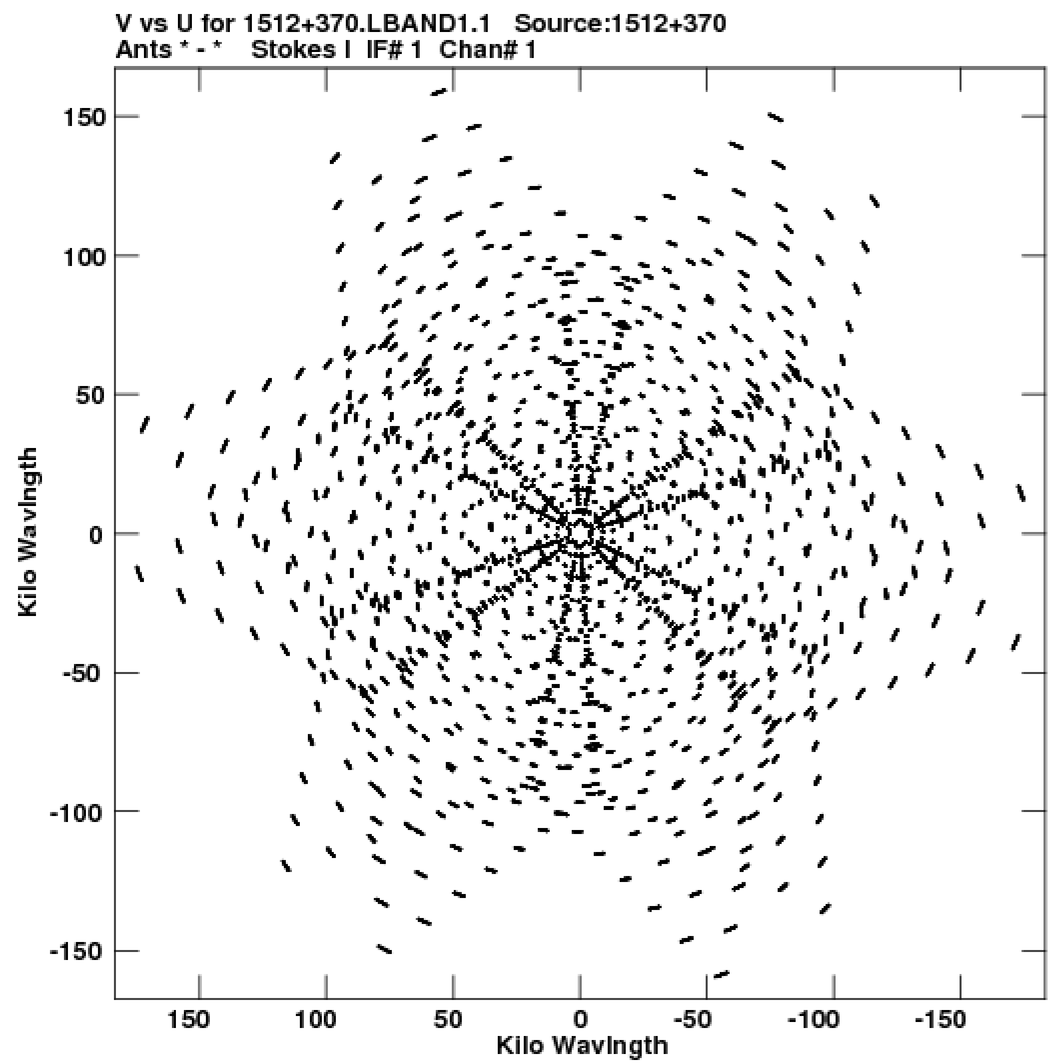
NRAO



		JPEG	Image Info	Links
			Stokes I Beam = 1.17 arcsec fov: R = 39.0 arcsec rms = 1.26 milliJy Download FITS image 397KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
		JPEG	Image Info	Links
1.66 GHz (L-band) Version: 2009 Nov 01	FITS 726KBytes		Stokes I Beam = 14.8 arcsec fov: R = 10.5 arcmin rms = 421. microJy Download FITS image 489KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
Basic Info	UV Info	JPEG	Image Info	Links
15h14m 0.0s 36d50'51" (J2000 image center) 1993 MAY 6 4.89 GHz (C-band) Version: 2007 Nov 17	Full polarization B/B configuration 27 antennas subarray 1 Calibrated uv-FITS 8.1MBytes		Stokes I Beam = 1.27 arcsec fov: R = 2.83 arcmin rms = 89.9 microJy Download FITS image 4.1MBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
Basic Info	UV Info	JPEG	Image Info	Links

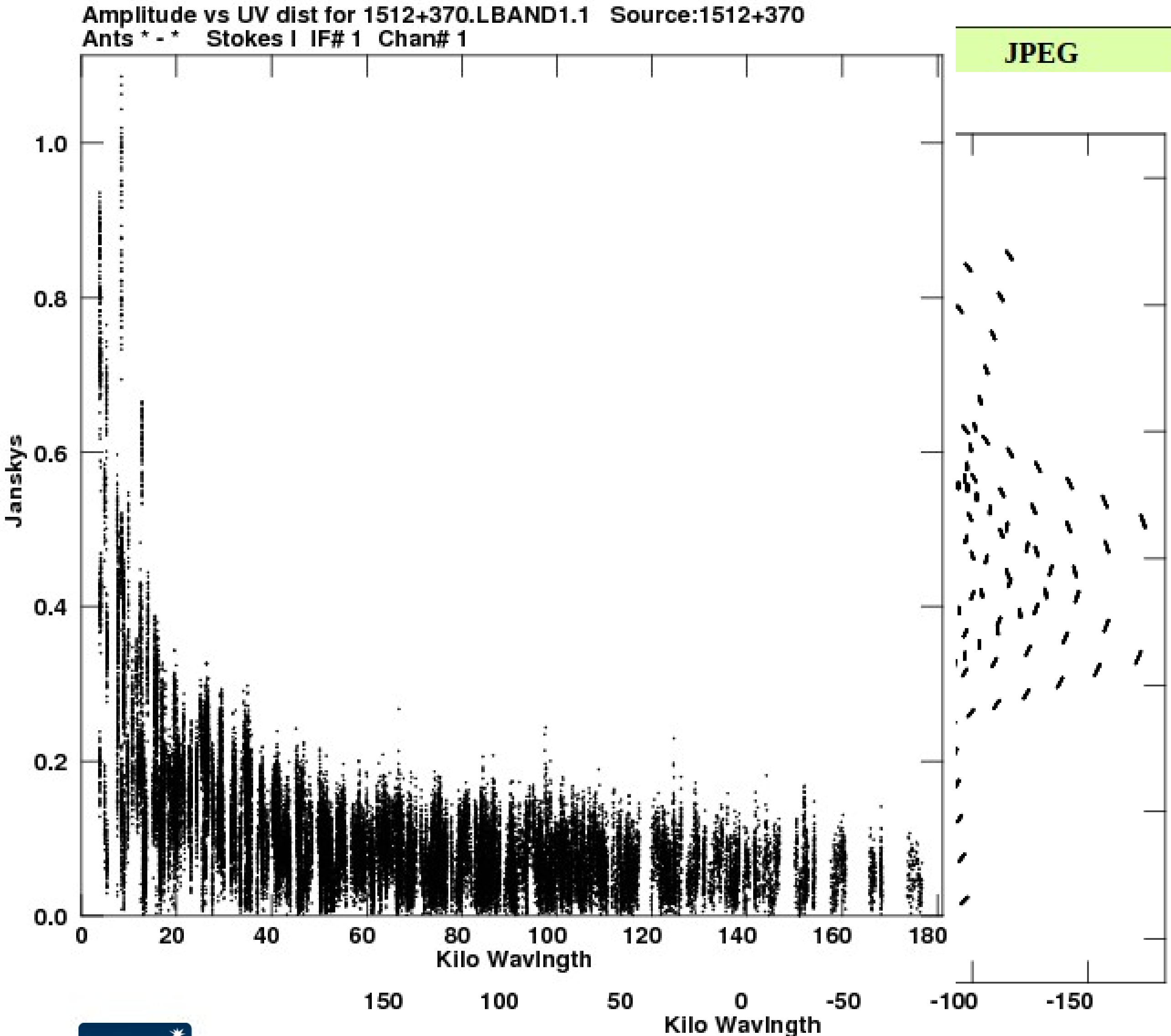
!

NRAO



JPEG	Image Info	Links
	Stokes I Beam = 1.17 arcsec fov: R = 39.0 arcsec rms = 1.26 milliJy Download FITS image 397KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
	Image Info	**Links**
	Stokes I Beam = 14.8 arcsec fov: R = 10.5 arcmin rms = 421. microJy Download FITS image 489KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
	Image Info	**Links**
	Stokes I Beam = 1.27 arcsec fov: R = 2.83 arcmin rms = 89.9 microJy Download FITS image 4.1MBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data

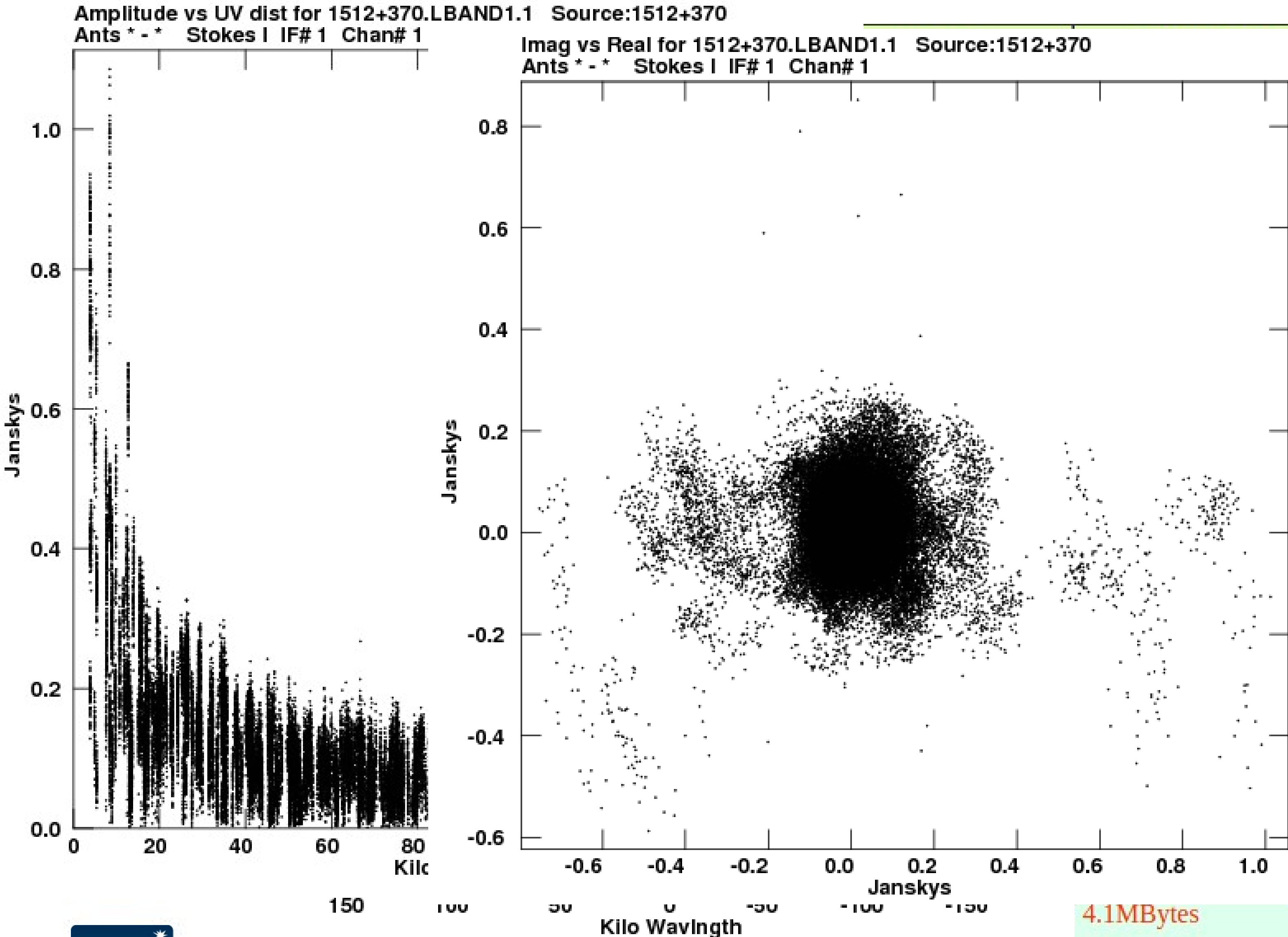
Basic Info	UV Info	JPEG	Image Info	Links



JPEG	Image Info	Links
	Stokes I Beam = 1.17 arcsec fov: R = 39.0 arcsec rms = 1.26 milliJy Download FITS image 397KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
	Image Info	**Links**
	Stokes I Beam = 14.8 arcsec fov: R = 10.5 arcmin rms = 421. microJy Download FITS image 489KBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
	Image Info	**Links**
	Stokes I Beam = 1.27 arcsec fov: R = 2.83 arcmin rms = 89.9 microJy Download FITS image 4.1MBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data

Basic Info	UV Info	JPEG	Image Info	Links

NRAO



Basic Info	UV Info	JPEG	Image Info	Links
				Links
			ge	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
				Links
			ge	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data
				Links
			ge 4.1MBytes	General data log u-v data coverage Re-Im data plot visibility plot AIPS pipeline runs Original u-v data

In summary:

- In practice pipelines (here) are similar in goal, but each which slightly different input and final data product
- Ask observatories for pipelines!
  → Documentation, use, pro's and con's…

- For end users/observers:
  - Does it do what I need it to do?
  - Are the data products acceptable?
  - What needs to be done further for my particular science goal?

NRAO

Design your own!

- Typically starts out as digitally edited "log" to reduce repetitive typing
- Making it more flexible, e.g. by defining the gain calibrator name as a variable
- Grabbing information from the data, e.g. number of channels to do edge flagging
- Calculations, e.g. the pixel size from the maximum baseline in wavelengths

→ sequence of instructions that needs less fiddling when applied to new data sets

Design your own!

- What should it be able to do?
  - Same observations done many times
    hard-coded copy/paste…
  - Same setup per source, many sources
    simple script (variables change)
  - Same source per setup, many setups
    complications… (frequency effects)
  - Everything including array configuration
    maximum flexibility!

Design your own!

- What is the starting point?
    - "Raw" data from the archive
      → modify the observatory pipeline?
    - Observatory pipeline products
    - Reference images
    - Other (manual?) processing
- What is the anticipated final product?
    - Determine success (versus "fail")

Design your own!

- Method:
  - Define sequence of steps In->Goal->Out
  - Decide what can be automated:
    - simple but repetitive actions
    - when keeping track is important
  - Where to put stopping points
    - pause for inspection checks
    - manually adjust/require input
- Formalize when each step is done & OK
  - go/no-go decisions
  - report results/plots in each step

NRAO

Perspectives and expectations

- Observatories want their users to publish and will help to achieve this:
  - provide pipelines and infrastructure for data reduction; you focus on the science
- New capabilities and insight, new heuristics:
  - pipelines remain under development
- Because of the general approach, you need to decide whether the product is sufficient
  - Products are on best effort basis

NRAO

Caution!

- Pipelines can only do what they are designed to do
- Pipelines cannot correct for bad observing conditions (rfi, weather, hardware, observing strategy/errors)
- Be aware of each pipeline's limitations (designer should expose these)

Caution!

- Pipelines are great, but typically are not tuned to your science requirements!
  - May do a good job, but you might do better with the proper investment

- Reprocessing might be an option (or not)

- Observations should **ideally** be set up with pipeline processing in mind
  - Probably best to keep it simple

NRAO

Aim

- Radio interferometry data sets are becoming complex and huge, and increasingly cumbersome to deal with
- Pipelines are already being widely used to process (radio interferometry) data
- Pipelines save a lot of time, but each have their pro's and con's