

Introduction to CASA

Anita Richards

AstroGrid / ALMA Regional Centre
JBCA, University of Manchester

freely plagiarising ESO and NRAO
presentations.....

CASA



The University of Manchester
Jodrell Bank
Observatory



CASA background

- Early 1990's: desire to replace AIPS (fortran)
 - Object-oriented c++ code easier to maintain
 - But aips++ never really became useful until:
- EVLA and ALMA demanded new methods
 - Focussed resources on specific problems
 - Tackled usability and real-world testers
- **C**ommon **A**stronomical **S**oftware **A**pplication
 - Python wrapper for many (not yet all) functions
 - 'Look and feel' closer to AIPS or MIRIAD!
 - Interactive or scripting interface
 - Extensive Cookbook/manual
 - Underlying c++ toolkit and CASACORE available

Who develops CASA and how

- Currently >20 people:
 - 14 NRAO+Virginia, 3 NAOJ, 2 ESO, 1 Calgary,
 - Fractions at Paris, ASTRON, ATNF
- Beta releases publicly available
 - Used seriously for BIMA/CARMA, VLA, ATCA
 - 'Official' testers use bug reporting system
 - European users feedback via ARC nodes/Zwaan
- Complements classic AIPS (view from here!)
 - Unlikely to handle all VLBI processing for a while
 - Users won't abandon familiar packages quickly
 - Interoperability is needed at crucial steps
 - ALBiUS (RadioNet) working on this

Measurement Set visibility data

- UNIX-like directory structure
- Manipulated via Measurement Equation matrix operations

```

jupiterallcal.split.ms
|-- ANTENNA
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock
|-- DATA_DESCRIPTION
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock
|-- FEED
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   |-- table.lock
|-- FIELD
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   |-- table.lock
|-- FLAG_CMD
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock
|-- HISTORY
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock
|-- OBSERVATION
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock
|-- POINTING
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.f1
|   |-- table.info
|   |-- table.lock
|-- POLARIZATION
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   |-- table.lock
|-- PROCESSOR
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock
|-- SOURCE
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   |-- table.lock
|-- SPECTRAL_WINDOW
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   |-- table.lock
|-- STATE
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   |-- table.lock

```

Measurement Equation

$$\underline{V}_{ij} = \mathbf{M}_{ij} \mathbf{B}_{ij} \mathbf{G}_{ij} \mathbf{D}_{ij} \int \mathbf{E}_{ij} \mathbf{P}_{ij} \mathbf{T}_{ij} \mathbf{F}_{ij} S \underline{I}_v(l,m) e^{-i2\pi(u_{ij}l + v_{ij}m)} dl dm + \underline{A}_{ij}$$

Vectors

\underline{V} visibility = $f(u,v)$

\underline{I} image to be calculated

\underline{A} additive baseline error

S (mapping \underline{I} to observer pol.)

l,m image plane coords

u,v Fourier plane coords

i,j telescope pair

Jones Matrices

M Multiplicative baseline error

B Bandpass response

G Generalised electronic gain

D term (pol. leakage)

E (antenna voltage pattern)

P Parallactic angle

T Tropospheric effects

F Faraday rotation

Measurement Equation

- *Hamaker Bregman & Sault 1996*
 - Decompose into individual calibration components e.g.
- $$\underline{V}_{ij}^{obs} = \mathbf{B}_{ij} \mathbf{G}_{ij} \mathbf{D}_{ij} \mathbf{P}_{ij} \mathbf{T}_{ij} \mathbf{F}_{ij} \underline{V}_{ij}^{ideal}$$
 - Linearise and solve by χ^2 minimization
- Other terms added as required
 - e.g. ζ Jones matrix (Jan Nordham)
 - (try saying it out loud....)

What's in the Measurement Set?

MAIN	Model, e.g.:	Corrected data	Flags
Original visibility data	<i>FT of image made from MS</i> <i>FT of supplied model image</i> <i>FT of calibrator flux density</i>	<i>Copy of visibilities with calibration tables applied</i> (Used in imaging but not calibration)	(Edits are stored here first; backup tables can be made and used to modify)

- Additional tables:
 - *Admin*: Antenna, Source etc.
 - *Processing*: calibration, flags, etc.

Visibility data handling

- **CASA can:**
 - Load FITS, VLA export, SDM (alpha release)
 - Load AN, FQ tabs and make source (field) list
 - Calibrate phase, amp, bandpass, polarization
 - Nice features like polynomial fitting
 - Apply multiple tables at once
 - Elaborate selection options - see examples
- **CASA can't (yet):**
 - Import FITS CL, FG etc. tables, D-terms
 - Calibrate <4 antennas for flux scaling
 - Solve delay, phase, rate VLBI combinations
 - Shift as in UVFIX
 - Flag using gains
 - Filter CLEAN model after imaging
 - WTMOD

Imaging

- Wide range of accessible choices
 - clean (also for mosaics)
 - widefield (facets or w-projection)
 - Antenna size & v -dependent primary beam corrections
 - Plan to allow multiple MS as input
 - Feasible to store & use spectral index, curvature etc. for MFS
- Current peculiarities
 - Searches entire cube for next peak each major cycle
 - Inefficient if many blank planes
 - Can't have mosaics *and* widefield
 - Have to edit in antenna diameter by hand
 - Can't turn off weighting
 - Viewer a bit painful

Overall impression

- Works well for most VLA
- Worth using for ATCA, CARMA etc. if AIPS unsuitable
- Not totally suitable for MERLIN yet
 - Inadequate control of weighting
 - Calibration flaky with v. small Nos antennas
 - Could be good for imaging calibrated, weighted data
 - **Great potential for MFS and wide field/mosaicing**
 - **Hope for field-based calibration implementation**
- Will be excellent for ALMA
 - See Almasimmos task and future developments
- And you can invoke any Python functions (including accessing the rest of your system!)

Feedback

- Can you swap extension tables between MS
 - what is specific to time, data shape etc...
 - and can you copy cal from L to R etc?
- Flagging slow because writes to visibilities?
 - and can you change extendflags type mid-flagging?
- How access CC (analysis, filtering for model)?
- Combine different data shapes in imaging?
 - can you get at *gridded* uv data?
- Does setjy have to be so slow for point model?
- Can you expand all params at one go in tasks?
- Developer information
 - e.g. combine aips++ tools for new imager

Feedback

- Need recipes to avoid data bloat in calibration
- Save us from setting task name to param name!
 - & field name not present message could be clearer
- Can you add subplots with different axes?
 - useful for interactive flagging
 - or open more than one viewer
 - time axis labelling more flexible
- Want pager-type control over viewing inputs
- Consistent save/restore in scripts/interactively
- Matrix-type listing (as in AIPS LISTR)
 - e.g. to spot anomalous visibilities
- 'I won't stop using aips++ any time soon
 - they call it Stockholm syndrome' (*Ian Heywood*)