Introduction to CASA

Anita Richards
with thanks to Dirk Petry (ESO) and the rest of the CASA teams
Common Astronomy Software Applications

- Original goal (\texttt{aips++}, 1990s):
  - To provide data reduction tools for radio interferometry, single dish and imaging generally
- Refocussed (and funded) in 2003 to be the ALMA and EVLA analysis package
  - RadioNet ALBIUS supports additional development
    - Other applications use underlying libraries
  - GNU Public License release in 2009

User interface, higher-level analysis routines, viewers
= \texttt{casa non-core} (Python wrappers)

General physical and astronomical utilities, infrastructure
= \texttt{casacore} (mostly c++)
CASA developers meeting 2010
CASA Architecture

- **Data structure**
  - Tables: Measurement set, caltables, images
- **Data import/export facilities**
  - FITS, Measurement Set, SDM, VLA export format
- **Tools for data access, display and editing**
  - Read/write between data formats, viewers
- **Tools for science analysis**
  - Based on Measurement Equation & related libraries
  - User-friendly 'task' interface
- **Programmable command line interface**
  - Scripting, full (i)Python functionality
- **Documentation**
  - Includes *Cookbook* for astronomers
Libraries use Measurement Equation

\[ V_{ij} = M_{ij}B_{ij}G_{ij}D_{ij} \int E_{ij}P_{ij}T_{ij}F_{ij}SI_{ij} (l,m)e^{-i2\pi(u_{ij}l+v_{ij}m)}dldm + \Delta_{ij} \]

**Vectors**
- Visibility = \( f(u, v) \)
- Image to be calculated
- Additive baseline error

** Scalars**
- \( S \) (mapping \( I \) to observer pol.)
- \( l,m \) image plane coords
- \( u,v \) Fourier plane coords
- \( i,j \) telescope pair

**Jones Matrices**
- Multiplicative baseline error
- Bandpass response
- Generalised electronic gain
- Term (pol. leakage)
- (antenna voltage pattern)
- Parallactic angle
- Tropospheric effects
- Faraday rotation

**Methods**
- Starting point
- The goal

**Hazards**
Using the Measurement Equation

- **Hamaker, Bregman & Sault 1996**
  - Decompose into individual calibration components e.g.
  \[ V_{ij}^{\text{obs}} = B_{ij}G_{ij}D_{ij}P_{ij}T_{ij}F_{ij}V_{ij}^{\text{ideal}} \]
  - Linearise and solve by $\chi^2$ minimization

- Same principles as any gain calibration
- Other terms added as required
  - e.g. $\zeta$ Jones matrix (© Jan Nordham)
- Visibility data are stored in Measurement Sets
  - Accessible directories of tables
What's in the Measurement Set?

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

- Additional tables:
  - **Admin**: Antenna, Source etc.
  - **Processing**: calibration, flags, etc.
### CASA Design & Implementation

#### Legend:

- **Table Name**
- **Key defined in this table**
- **key definition method**
- **[referenced by]**
- **[referenced keys]**
- **(optional)**
- **reference to table outside the MS definition**

| Level 1: Tables not referenced by others |
| Level 2: Tables referenced by level 1 |
| Level 3: Tables referenced by level 2 |

#### Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Key defined in this table</th>
<th>Key definition method</th>
<th>[referenced by]</th>
<th>[referenced keys]</th>
<th>reference to table outside the MS definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>none</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ANTEenna</td>
<td>ANTENNA_ID</td>
<td>row number</td>
<td>MAIN</td>
<td>(ORBIT_ID)</td>
<td>(PHASED_ARRAY_ID)</td>
</tr>
<tr>
<td>FEED</td>
<td>FEED_ID</td>
<td>explicit</td>
<td>MAIN</td>
<td>FREQ_OFFSET</td>
<td>(PHASED_FE)</td>
</tr>
<tr>
<td>DATA_DESCRIPTION</td>
<td>DATA_DESC_ID</td>
<td>row number</td>
<td>MAIN</td>
<td>SPW_ID</td>
<td>POLARIZATION_ID</td>
</tr>
<tr>
<td>PROCESSOR</td>
<td>PROCESSOR_ID</td>
<td>row number</td>
<td>MAIN</td>
<td>TYPE_ID</td>
<td>(MODE_ID)</td>
</tr>
<tr>
<td>FIELD</td>
<td>FIELD_ID</td>
<td>row number</td>
<td>MAIN</td>
<td>SOURCE_ID</td>
<td>(EPHEMERIS_ID)</td>
</tr>
<tr>
<td>OBSERVATION</td>
<td>OBSERVATION_ID</td>
<td>row number</td>
<td>MAIN</td>
<td>HISTORY</td>
<td>none</td>
</tr>
<tr>
<td>STATE</td>
<td>STATE_ID</td>
<td>row number</td>
<td>MAIN</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>SPW</td>
<td>SPW_ID</td>
<td>row number</td>
<td>DATA_DESCRIPTION</td>
<td>(RECEIVER_ID)</td>
<td>(DOPPLER_ID)</td>
</tr>
</tbody>
</table>

#### Attributes

- **ANTENNA_ID**
- **FEED_ID**
- **DATA_DESC_ID**
- **PROCESSOR_ID**
- **PHASE_ID**
- **FIELD_ID**
- **ARRAY_ID**
- **OBSERVATION_ID**
- **STATE_ID**
- **SPW_ID**
- **SOURCE_ID**
- **RECEIVER_ID**
- **DOPPLER_ID**
- **ASSOC_SPW_ID**

#### Level Structure

- **Level 1**: Tables not referenced by others
- **Level 2**: Tables referenced by level 1
- **Level 3**: Tables referenced by level 2

---

A M S Richards  MWA 2010  p8  Printed: 28/06/10
CASA special features

- Framework architecture of 17 tools can be bound to any scripting language

at – atmosphere library
ms – Measurement Set utilities
mp – Measurement Set Plotting, e.g. data (amp/phase) versus other quantities
cb – Calibration utilities
cp – Calibration solution plotting utilities
im – Imaging utilities
ia – Image analysis utilities
fg – flagging utilities
tb – Table utilities (selection, extraction, etc.)
me – Measures utilities
tp – table plot
vp – voltage patterns
qa – Quanta utilities
cs – Coordinate system utilities
pl – matplotlib functionality
sd – ASAP = ATNF Spectral Analysis Package (single-dish analysis)
sm – simulation
Measurement Set visibility data

- Directory of Tables
- **MAIN** table
  - One row per integration per baseline per spectral window
- Cells hold complex visibilities and weights
Measurement Set MAIN table

- Some of the columns per visibility
  - **Data**: Complex value for each of 4 correlations (LL RR LR RL) per spectral channel
Starting CASA

• See web links for downloads (or http://casa.nrao.edu)
  − Don't forget the Cookbook!
• Start by typing casapy
  − This starts the iPython environment
    • Interactive input to tasks in the xterm
    • Logger (see toolbar for display, export options)
  − Access to shell
    • Direct simple commands e.g. ls
    • Prefix any unix command with ! e.g. !more file
• Python
  − Take care with indentation
  − Case sensitive
  − Zero indexed (e.g. 27 antennas numbered 0~26)
    • Run any scripts or functions you want
Using CASA

- Use `inp taskname` to view inputs
  - Greyed parameters are expandable

```plaintext
CASA <37>: inp gaincal
--------> inp(gaincal)
# gaincal :: Determine temporal gains from calibrator observations
vis      = '3C277,1C,ms' # Name of input visibility file
caltab   = '' # Name of output gain table
field    = '' # Select field using field id(s) or field name(s)
spw      = '' # Select spectral window/channel
selectdata = False # Other data selection parameters
solint   = 'inf' # Solution interval: e.g., 'inf', '60s' (see help)
```
Using CASA

```python
CASA <38>: selectdata = True

CASA <39>: inp gaincal
----------- inp(gaincal)
# gaincal :: Determine temporal gains from calibrator observations
vis      = '3C277.1C.ms'       # Name of input visibility file
caltable = ''                 # Name of output gain calibration table
field    = ''                 # Select field using field id(s) or field name(s)
spw      = ''                 # Select spectral window/channel(s)
selectdata = True             # Other data selection parameters
# Select data based on time range
# Select data within uvrange (default units meters)
# Select data based on antenna/baseline
# Scan number range
# Optional complex data selection (ignore for now)
solint   = 'inf'             # Solution interval: e.g., 'inf', '60s' (see help)
```
Using CASA

• Simplest input to tasks is param=value
  - In this mode, variables are global
    • solint='1min' will appear in all tasks until reset
  - default(gaincal) resets default values
  - tget gaincal restores last successful execution
  - saveinputs(gaincal,'gctry1') saves inputs at any stage
  - execfile('gctry1') restores
    • gctry1 is a text file, view using e.g. !more gctry1

• Help('gaincal') for more details
  - Use the Cookbook for fuller examples
Running tasks

• In interactive mode
  – Just type e.g. gaincal
  – Tasks are normally run sequentially per session
  – See the logger for progress

• Assign measurements to variables
  – e.g. noise_target = imstat()
    • Python syntax examples in scripts or cookbook
    • rms_target=noise_target['rms'][0]

• Beware re-assigning/mistyping task params
  – molint = '1sin' won't give an error
  – calmode = 'delay' does show up in red
CASA functionality

- CASA converts between FITS and MS
  - Apply calibration etc. first
  - Default is not to overwrite
    - Except when continuing `clean`
- CASA MS and images are directories
  - Move, delete, rename etc. using shell commands
    - See Cookbook for utilities in scripting e.g. `rmtables`
- Logfiles:
  - `ipython.log` records commandline
    - Per window, but will be overwritten in new session!
  - `casapy.log` records task messages
    - Renamed by date/time when a new session starts
  - History table attached to data
CASA, the shell and Python

• Can use any shell command inside CASA via “!” e.g. !emacs ipython.log

• To run script inside CASA:
  execfile('my.py')

• Can use tabcomplete, auto() etc.
  - uparrow to recall previous commands
  - Indentation matters, but more forgiving than pure python

• Zero indexed

• ^D or exit to exit

• ^C or shell kill to stop a task
  - Occasional lock problems; exit and/or check for zombie processes
Time jargon

Total integration time = 456357 seconds
Observed from 15-Apr-1995/17:13:58.0 to 20-Apr-1995/UTC

<table>
<thead>
<tr>
<th>Timerange (UTC)</th>
<th>Scan</th>
<th>Field Name</th>
<th>nVis</th>
<th>Int(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:13:58.0 - 17:28:38.0</td>
<td>1</td>
<td>03C286</td>
<td>1665</td>
<td>7.99</td>
</tr>
<tr>
<td>17:29:38.0 - 18:29:30.0</td>
<td>2</td>
<td>1OQ208</td>
<td>6750</td>
<td>7.99</td>
</tr>
<tr>
<td>17:07:38.0 - 17:09:54.0</td>
<td>8</td>
<td>10 1300+580</td>
<td>270</td>
<td>7.99</td>
</tr>
<tr>
<td>17:10:37.0 - 17:17:49.0</td>
<td>9</td>
<td>11 3C277.1</td>
<td>825</td>
<td>7.99</td>
</tr>
<tr>
<td>17:18:36.0 - 17:19:56.0</td>
<td>10</td>
<td>10 1300+580</td>
<td>165</td>
<td>7.99</td>
</tr>
<tr>
<td>17:20:35.0 - 17:27:55.0</td>
<td>11</td>
<td>11 3C277.1</td>
<td>840</td>
<td>7.99</td>
</tr>
<tr>
<td>17:28:42.0 - 17:29:54.0</td>
<td>12</td>
<td>10 1300+580</td>
<td>150</td>
<td>7.99</td>
</tr>
</tbody>
</table>

- **Time on all sources**
- **Span** of observations (might be gaps)
- Flux scale/polarisation calibration scans
- Alternate phase-ref/target scans
- Single integration time

- Estimate hour angle coverage
- An integration is the shortest averaging time in correlated data
- A scan is usually the time between source changes
  - The phase-ref/target cycle should be less than the atmospheric coherence time
Tutorials

- CASA: Calibration and imaging
  - MERLIN and EVLA data
  - Continuum, spectral lines and polarization
  - Scripting and image analysis
  - Simulations (for ALMA in this example)

- AIPS: Combining arrays and VLBI
  - EVN (+ MERLIN)

- Additional CASA material if you have time:
  - Mosaicing, wide-field imaging, analysis
    - VLA, EVLA continuum, ATCA HI
    - mm-wave data: BIMA, SMA and CARMA
  - Work at your own pace
    - Experiment
    - Make sure you understand what you are doing
Typical CASA flowchart

1. Flagging and Calibration
Typical CASA flowchart

2. Imaging and analysis

- applycal
- Calibrated Data
- imaging
- image cube
- numerical analysis
- viewing, plotting

Publication-ready plots and numerical results
Science view

- Approx flux and bandpass scaling pre-applied
- Set known fluxes (allowing for resolution if necessary)
- Phase-cal calibration sources
  - A&P self-cal for phase ref, bp cal source
  - Derive bandpass cal if required
  - Fluxscale if required
- Solve for pol. leakage, usually with phase ref source, apply
- Correct pol angle, usually with 3C286, apply
- Apply phase-ref solutions and image target
  - Phase self-cal if enough SNR
  - Image target, A&P self-cal if enough SNR
- Final target imaging
- Inspect data and solutions regularly. Flag if required.

Calibration is incremental. Apply as required.
Self-Calibration minimising data volume

Data for selfcal

Clean_1

Phase self-cal

Clean_final

Applycal

Clean_2

new model

_overwrites model

Clean_1

Data for selfcal

new model

Clearcal (re-) correct

Clean_final

Applycal

a&p self-cal

Apply original phase cal & new amp cal with incremental phase cal