Introduction to CASA

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

Common Astronomy Software Applications

- Original goal (`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
= *casa non-core* **(Python wrappers)**

↕

General physical and astronomical utilities, infrastructure
= *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

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

Vectors

$\underline{V}$ isibility = $f(u,v)$ — Starting point

$\underline{I}$ mage — The goal

$\underline{A}$ dditive baseline error

Scalars

Methods

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

l,m image plane coords

u,v Fourier plane coords

i,j telescope pair

Jones Matrices

Hazards

Multiplicative baseline error

Bandpass response

Generalised electronic gain

Dterm (pol. leakage)

E (antenna voltage pattern)

Parallactic angle

Tropospheric effects

Faraday rotation

Using the 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
- Same principles as any gain calibration
- Other terms added as required
  - e.g. ζ Jones matrix (© Jan Nordham)
- Visibility data are stored in Measurement Sets
  - Accessible directories of tables

What's in the Measurement Set?

MAIN	Model, e.g.:	Corrected data	Flags
Original visibility data	*FT of image made from MS* *FT of supplied model image* *FT of calibrator flux density*	*Copy of visibilities with calibration tables applied* (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.

CASA Design & Implementation

Table Name	Key defined in this table	Key definition method	Referenced by	Referenced keys
MAIN	none	-	-	ANTENNA_ID FEED_ID DATA_DESC_ID PROCESSOR_ID (PHASE_ID) FIELD_ID (PULSAR_GATE_ID) ARRAY_ID OBSERVATION_ID STATE_ID
(FREQ_OFFSET)	none	-	-	ANTENNA_ID FEED_ID SPW_ID
(SYSCAL)	none	-	-	ANTENNA_ID FEED_ID SPW_ID
POINTING	none	-	-	ANTENNA_ID POINTING_MODEL_ID
(WEATHER)	none	-	-	ANTENNA_ID
HISTORY	none	-	-	OBSERVATION_ID OBJECT_ID
FLAG_CMD	none	-	-	none
ANTENNA	ANTENNA_ID	row number	MAIN FEED FREQ_OFFSET POINTING SYSCAL WEATHER	(ORBIT_ID) (PHASED_ARRAY_ID)
FEED	FEED_ID	explicit	MAIN FREQ_OFFSET SYSCAL	ANTENNA_ID SPW_ID BEAM_ID (PHASED_FE
DATA_DESCRIPTION	DATA_DESC_ID	row number	MAIN	SPW_ID POLARIZATION_ID (LAG_ID)
PROCESSOR	PROCESSOR_ID	row number	MAIN	TYPE_ID MODE_ID (PASS_ID)
FIELD	FIELD_ID	row number	MAIN	SOURCE_ID (EPHEMERIS_ID)
OBSERVATION	OBSERVATION_ID	row number	MAIN HISTORY	none
STATE	STATE_ID	row number	MAIN	none
SPW	SPW_ID	row number	DATA_DESCRIPTION FEED FREQ_OFFSET SOURCE SYSCAL	(RECEIVER_ID) (DOPPLER_ID) (ASSOC_SPW_ID)
POLARIZATION	POLARIZATION_ID	row number	DATA_DESCRIPTION	none
(SOURCE)	SOURCE_ID	explicit	(DOPPLER) FIELD	SPW_ID (PULSAR_ID)
(DOPPLER)	DOPPLER_ID	explicit	SPW_ID	SOURCE_ID TRANSITION_ID

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

CASA special features

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

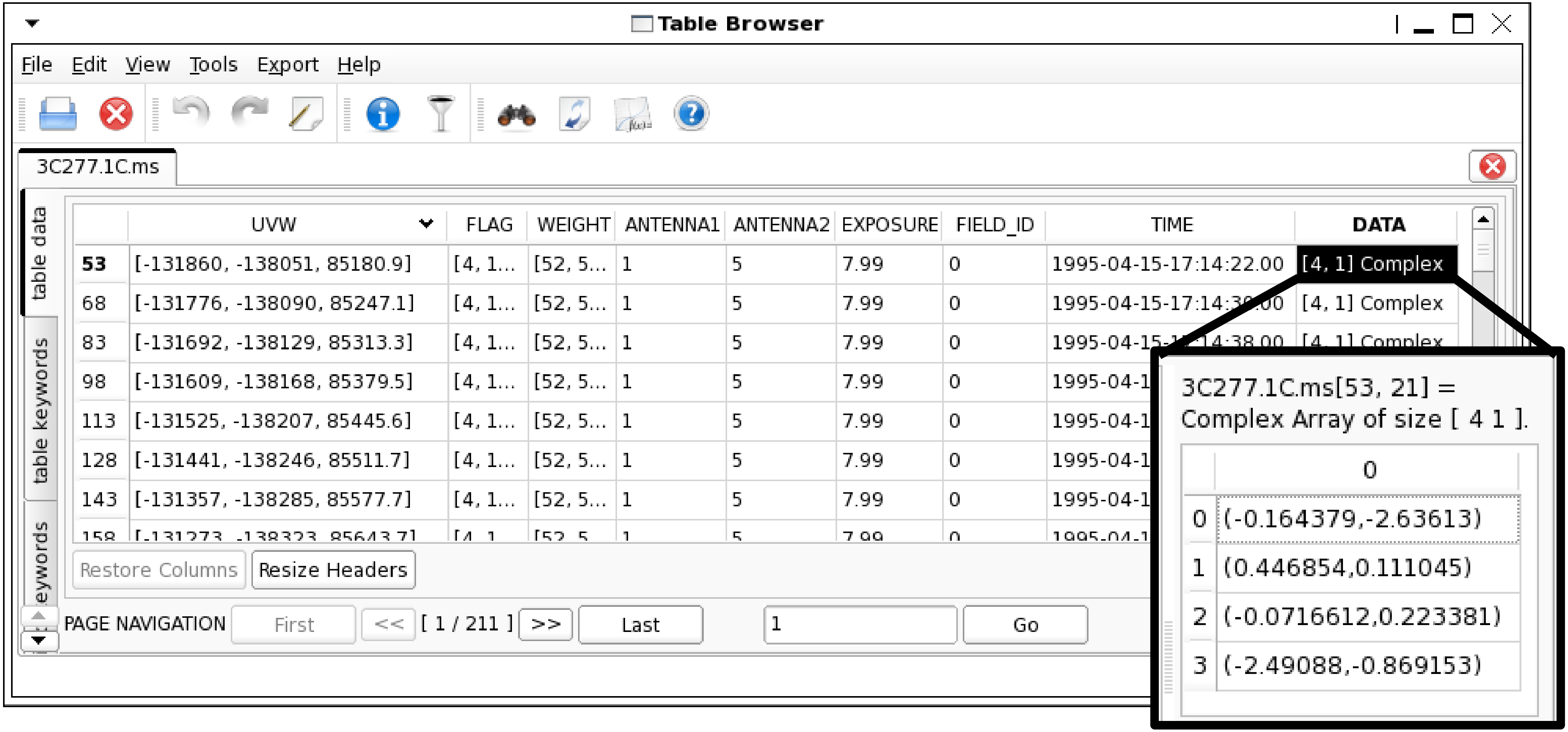
at – atmosphere library
ms – Measurement Set utilities
mp – Measurement Set Plotting, e.g. data (amp/phase) versus other quantities
cb – Calibration utilities
cp – Calibration solution plotting utilities
im – Imaging utilities
ia – Image analysis utilities
fg – flagging utilities
tb – Table utilities (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

```
jupiterallcal.split.ms
|-- ANTENNA
|    |-- table.dat
|    |-- table.f0
|    |-- table.info
|    `-- table.lock
|-- DATA_DESCRIPTIO
|    |-- 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
|-- table.dat
|-- table.f0
|-- table.f1
|-- table.f2
|-- table.f2_TSM1
|-- table.f3
|-- table.f3_TSM1
|-- table.f4
|-- table.f5
|-- table.f6
|-- table.f6_TSM0
|-- table.f7
|-- table.f7_TSM1
|-- table.f8
|-- table.f8_TSM1
|-- table.info
`-- table.lock
```

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

xterm

```
CASA <37>: 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/channels
selectdata          =      False        #  Other data selection
                                        #   parameters
solint              =      'inf'        #  Solution interval: egs.
                                        #   'inf', '60s' (see help)
```

Using CASA

xterm

```
CASA <38>: selectdata = True

CASA <39>: inp gaincal
---------> inp(gaincal)
#  gaincal :: Determine temporal gains from calibrator observations
vis                 =  '3C277.1C.ms'     #  Nome 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/channels
selectdata          =        True        #  Other data selection
                                         #   parameters
     timerange      =           ''       #  Select data based on time
                                         #   range
     uvrange        =           ''       #  Select data within uvrange
                                         #   (default units meters)
     antenna        =           ''       #  Select data based on
                                         #   antenna/baseline
     scan           =           ''       #  Scan number range
     msselect       =           ''       #  Optional complex data
                                         #   selection (ignore for now)

solint              =       'inf'        #  Solution interval: egs.
                                         #   '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

EUROPEAN ARC
ALMA Regional Centre || UK

CASA

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 utilties 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 attatched 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)

Timerange (UTC)	Scan	FldId	FieldName	nVis	Int(s)
17:13:58.0 - 17:28:38.0	1	0	3C286	1665	7.99
17:29:38.0 - 18:29:30.0	2	1	OQ208	6750	7.99
.....					
17:07:38.0 - 17:09:54.0	8	10	1300+580	270	7.99
17:10:37.0 - 17:17:49.0	9	11	3C277.1	825	7.99
17:18:36.0 - 17:19:56.0	10	10	1300+580	165	7.99
17:20:35.0 - 17:27:55.0	11	11	3C277.1	840	7.99
17:28:42.0 - 17:29:54.0	12	10	1300+580	150	7.99

- Time on all sources
- Span of observations (might be gaps)
- Flux scale/polarisation calibration scans

- Alternate phase-ref/target scans
- Single integration time

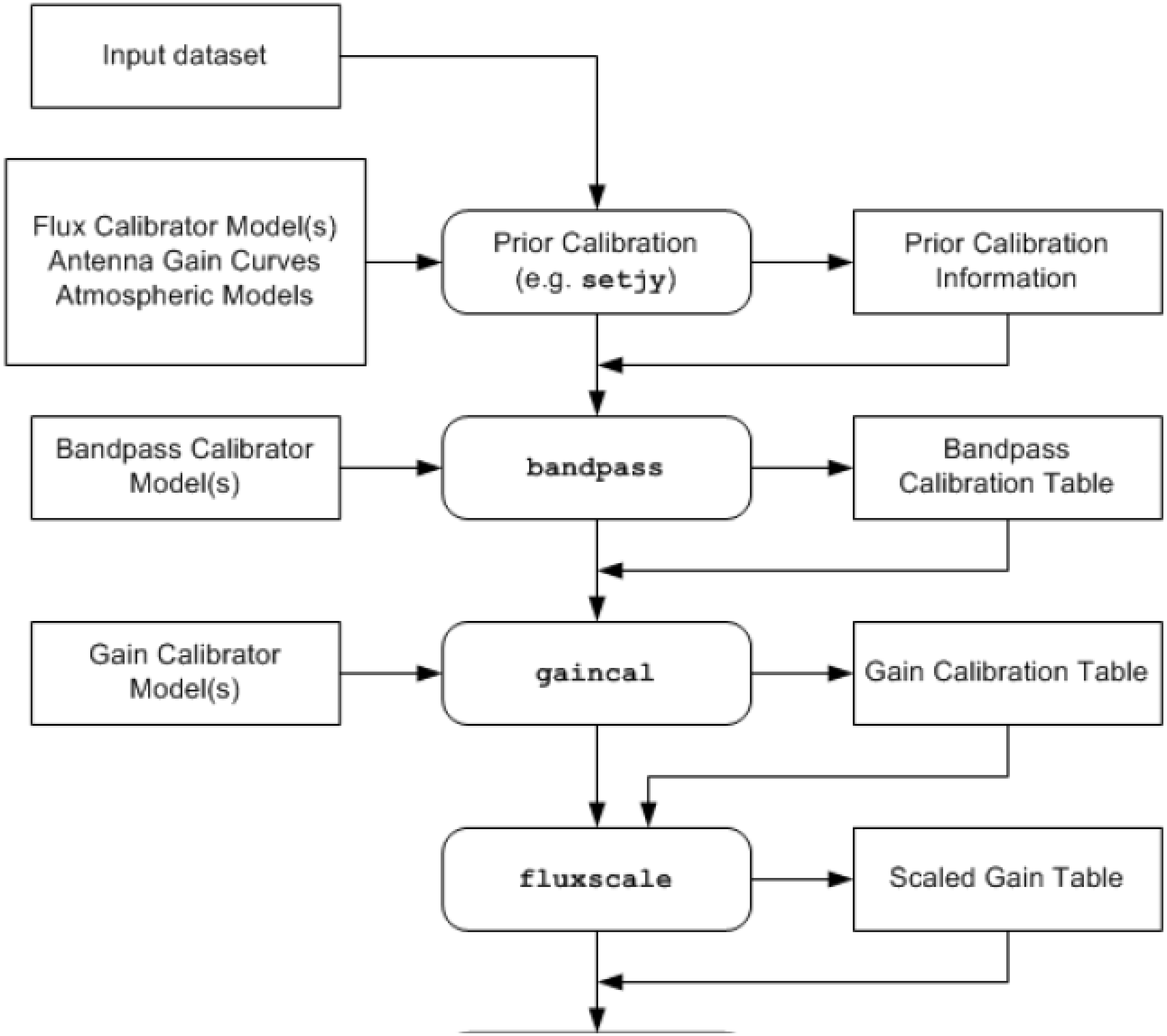
- Estimate hour angle coverage
- An integration is the shortest averaging time in correlated data
- A scan is usually the time between source changes
  - The phase-ref/target cycle should be less than the atmospheric coherence time

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

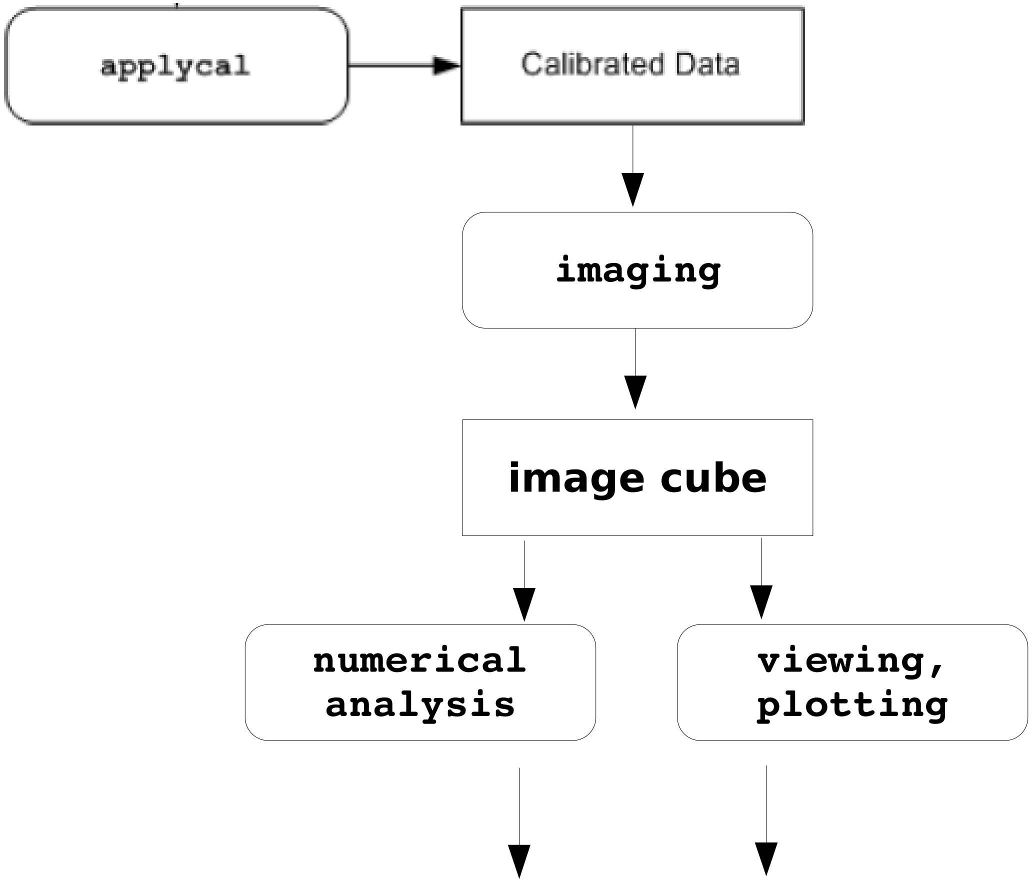
Typical CASA flowchart

1. Flagging and Calibration



Typical CASA flowchart

2. Imaging and analysis



publication-ready plots and numerical results

Science view

Calibration is incremental. Apply as required.

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

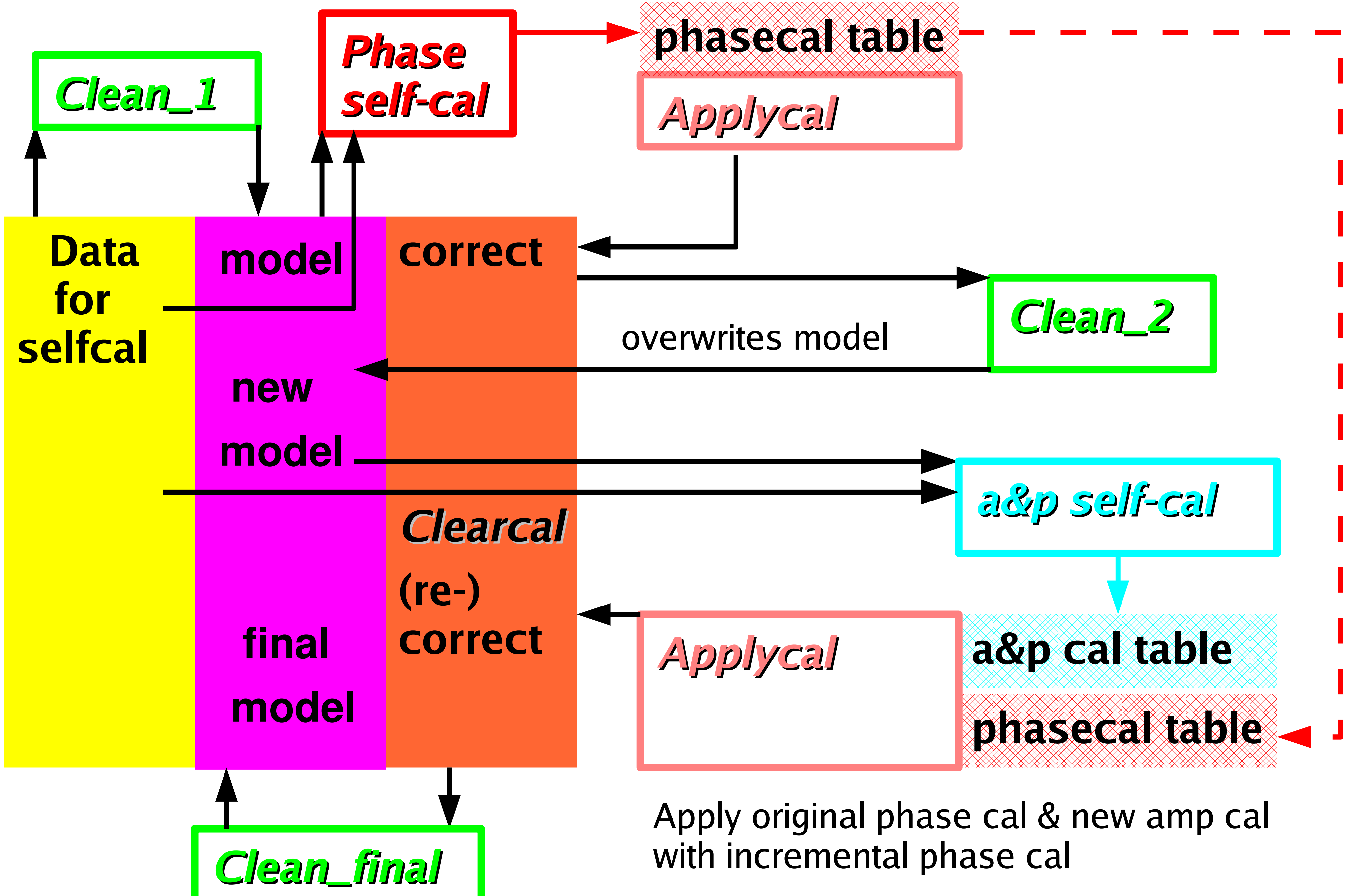
Phase self-cal if enough SNR

Image target, A&P self-cal if enough SNR

Final target imaging

Inspect data and solutions regularly. Flag if required.

Self-Calibration minimising data volume



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