ALMA Pipeline Heuristics

Frederic Boone - LERMA
Lindsey Davis - NRAO
John Lightfoot - UKATC
Dirk Muders - MPIfR
Christine Wilson - McMaster
Friedrich Wyrowski - MPIfR
Luis Zapata - MPIfR

The Mission

- Automatic reduction of ALMA data; single field interferometry, mosaics, single-dish

  Useful  publishable

  Quick look, system health

- Observers must trust the results

  Transparent
  verifiable

- Easy to use, configure, modify
- Ready for ‘early science’

The Tools

- Casapy - Python binding of CASA tools
- Python
- numpy - array operations
- Matplotlib - display
- Others?

Need to do

- Automatically detect and flag bad data
- Find the ‘best’ reduction method -

  Bandpass Calibration:
  channel
  polynomial fit (degree?, bandpass edges?)
  quality of solution?

  Phase Calibration:
  interpolate / spline fit / combine spectral windows

- Find ‘best’ way to calculate result; e.g clean map - code it

Design

- ‘Recipe’ specifies a series of reduction ‘stages’
- Stage can flag data, search for the ‘best’calibration method, calculate a result

Stage sequence

Improve flagging
Improve calibration method

- Each stage is an object. The ‘bandpass calibration’ and ‘phase calibration’ are objects. O-O encapsulation helps keep code manageable

‘Flagging’ stage

- Data ‘view’
  - -Direct access to MeasurementSet - TaQL
  - -Modified raw data e.g. median across channels for each baseline/timestamp
  - -Processed data e.g. antenna based gain amplitudes for each timestamp
  - -Calibration results, as would be applied to data
  - -Metadata; Tsys, water vapour column
- Flagging
  - -Flag specific data, e.g. autocorrelations
  - -Calculate statistics of view, flag outliers
  - -Detect bandpass edges
- Display
  - -Image
  - -line plot
  - -‘before’ and ‘after’ flagging display, data colour keyed to reason for flagging

‘Best’ Method Stage

- Currently prototype for bandpass calibration

Scattergun appoach - try a variety of methods - test - adopt best.
Variations in: G_t (phasing up of data before calculating bandpass)
Channel calibration
Polynomial fit calibration - poly degree
Test by calculating a ‘figure of merit’ for each calibration method:
Calibrate a test field (different to the bandpass data)
Measure flatness of result
Adopt method producing lowest figure of merit for future calibrations.

Result Stage

- For example, a clean map/cube

  Specify the Python class to calculate the clean image

  Python objects to supply the bandpass calibration and the phase calibration are passed as parameters

  Basically a ‘canned’ method for calculating the result - library of these

Status

- Recipes for VLA and Plateau de Bure data
- Recipe for ‘quick look’
- Prototype mosaic recipe
- Next, implement selection of ‘best’ method for phase calibration