The *CASA* software package

Dirk Petry (ESO/EU ARC), March 2018

Outline

→ What is *CASA?*

→ Who develops *CASA*?

→ *CASA* design and implementation

→ *CASA* status and release plans

→ Typical ALMA interferometric and SD analysis/combination workflow

CASA main features

- *CASA* = **Common Astronomy Software Applications**
- **Development started in the 90s as the next generation of AIPS**
- **Re-focussed in 2003 to be *the ALMA/EVLA analysis package***
- **Has the intention to be a *general software package to reduce both interferometer and single-dish data***
- **Internally consists of two parts:**

User interface, higher-level analysis routines, GUIs
= *casa non-core*

General physical and astronomical utilities, infrastructure
= *casacore*

- **Implements the "Measurement Equation" (Hamaker, Bregman & Sault 1996)**
- **Internal data format is the "Measurement Set" (Kemball & Wieringa 2000)**
- **more than 1.5 Million lines of code (mostly C++)**
- **In public release under GNU Public License since December 2009**

CASA main features

Since release 5.1, CASA releases include the

Science Pipelines

for both ALMA and the VLA (share common base)

Needed
a) by the observatories to *create* the standard calibration and imaging products

b) by the observatory users to *restore* calibrated data from the raw data based on delivered calibration tables

Can also be used to perform sophisticated custom calibration and imaging.

For more information, see
https://almascience.eso.org/processing/science-pipeline
https://science.nrao.edu/facilities/vla/data-processing/pipeline

Who develops CASA?

CASA non-core: the part geared to ALMA and VLA, developed mostly with
ALMA and VLA funding at NRAO, ESO, NAOJ, ASIAA,
Recently, an MoU on collaboration was signed with the SKA
http://casa.nrao.edu, repository on github

User interface, higher-level analysis routines, GUIs
= *casa non-core*

General physical and astronomical utilities, infrastructure
= *casacore*

Casacore: the general-use part developed by a larger consortium of institutes:
ASTRON, CSIRO, SKA SA, and again NRAO, ESO, NAOJ
https://github.com/casacore/casacore

CASA design and implementation

Overall architecture:

1) A data structure

2) A set of data import/export facilities

3) A set of tools for data access, display, and editing

4) A set of tools for science analysis

5) A set of high-level analysis procedures ("tasks")

6) A programmable command line interface with scripting

7) Documentation

CASA design and implementation

Overall architecture:

1) A data structure

Tables: Images, Caltables, and the *Measurement Set* (MS)

2) A set of data import/export facilities

the so-called *fillers*: (ASDM, UVFITS, FITS-IDI, VLA archive) → MS,   FITS → Image

3) A set of tools for data access, display, and editing

tools to load/write data into/from casacore data types,

table browser, viewer, and x/y plotter

4) A set of tools for science analysis

Calibration model built around the *Measurement Equation* (developed in 1996),

a toolkit for radio astronomical calibration, imaging, and simulation

5) A set of high-level analysis procedures ("tasks")

user-friendly implementations of the solutions for all common analysis problems

6) A programmable command line interface with scripting

Python 2.7 (augmented by `IPython`) gives a MATLAB-like interactive language

7) Documentation

extensive documentation through help commands and online **CASADocs**

and **CASA Guides**. See **http://casa.nrao.edu/**

CASA design and implementation

CASA data structure:

the *Measurement Set* (MS)

    MS = table for radio telescope data + auxiliary sub-tables

     - designed to store both interferometry and single-dish data

     - supports (in principle) any setup of radio telescopes

     - *relational database system with fixed structure:*
        a main table with 12 required sub-tables + several optional ones

     - *MS v2.0* described in http://casa.nrao.edu/Memos/229.html

     - *in CASA implemented as CASA Tables*

     - manipulate an MS with the `ms` and the `tb` tool or with `browsetable()`

     - comfortable access to MS metadata through the `msmd` tool

CASA design and implementation

CASA data structure:

the *Measurement Set* (MS)

MS = table for radio telescope data + auxiliary sub-tables

- data columns in the MS main table

DATA — **- the original visibilities (complex)**
(not a scratch column, always present for interf. data)

FLOAT_DATA — **- intended for SD amplitudes (real)**
(CASA SD code uses DATA column!)

CORRECTED_DATA — **- calibrated values obtained based on DATA**
(initialised with a copy of DATA)

MODEL_DATA — **- Fourier transform of a model image**
(initialised to 1)

- **MODEL_DATA can be virtual (controlled by parameter *usescratch*),
if *usescratch=False,* a virtual MODEL column is calculated on demand from
information in the MS header**

- **use tasks *delmod, setjy, clearcal, ft* to manipulate the models**

CASA design and implementation

CASA tools for science analysis:

A toolkit for interferometric imaging

Interferometric Imaging in CASA: Combinations of Major and Minor Cycle Algorithms

Imaging (Major Cycle):

1) Standard (no dir.-dep. effects, uv-grid sampling uses convolutional regridding)

2) with dir.-dep. effects:

a) W-term (image domain faceting, uv domain faceting, W projection)

b) PB correction (image domain, A projection)

c) Pointing Offset correction by phase gradient

d) Mosaicing (linear (separate) deconvolution,
joined deconv. of combined dirty images,
mosaicing by regridding all uv data onto one grid)

Deconvolution (Minor Cycle):

1) CLEAN (delta function model)

2) MS-CLEAN (blob model)

3) MSMFS CLEAN (model of blobs with polynomial spectrum)

4) MEM (maximum entropy method using prior image and delta function model)

see nice overview compiled by Urvashi Rau: https://safe.nrao.edu/wiki/bin/view/Software/AlgorithmList
(somewhat deprecated but a new page is still under development)

CASA design and implementation

CASA tools for science analysis:

A toolkit for single-dish imaging

Before CASA 5.0, SD processing in CASA was based on the
   Australian Telescope National Facility Spectral Analysis Package (ASAP)
   + the ScanTable data format
   (originated from CSIRO, adapted to ALMA by NAOJ)

Now with CASA 5.1 (end of 2017):
   - ASAP is no longer part of CASA
   - all SD functionality migrated to *use MS data format*:
      - avoids potential data format confusion and reduces learning costs
      - permits reuse of existing MS-related CASA infrastructure
      - unified processing for ALMA TP, Nobeyama, and ASTE data
   - uses NAOJ's high-performance array processing library "*Sakura*"
   - ALMA SD pipeline was migrated as well
      https://almascience.eso.org/processing/documents-and-tools/alma-science-pipeline-users-guide-casa-5-1.1
   - an ALMA usage example is documented in detail in
      The ALMA M100 Band 3 Single-Dish CASA guide
      https://casaguides.nrao.edu/index.php/M100_Band3_SingleDish_5.1

CASA design and implementation

A programmable command line interface with scripting

*Framework Architecture of 29 tools bound to **Python***

See output of *toolhelp*

The most often used tools for the standard user are:

```
cb : Calibration utilities
cs : Coordinate system utilities
ia : Image analysis utilities
im : Imaging utilities
me : Measures utilities
ms : MeasurementSet (MS) utilties
msmd: MS metadata accessors
rg : Region manipulation utilities
tb : Table utilities (selection, extraction, etc)
qa : Quanta utilities
sl : Spectral line import and search
sm : Simulation utilities
sd : Single dish utilities
vp : voltage pattern/primary beam utilities
```

A programmable command line interface with scripting

(continued)

Python (augmented by `IPython 5.1.0`)

Gives features such as

- tab completion
- autoparenthesis
- command line numbering
- access to OS, e.g.

    Lines starting with '!' go to the OS.

    `a = !ls *.py` to capture the output of '`ls *.py`'.

    `!cmd $myvar` expands Python var `myvar` for the shell, e.g.

    !ls $os.getenv("CASAPATH").split(' ')[0]"/data/alma/simmos/".
- history
- **execfile***(<script file name>)* to run a given Python script within CASA
- comfortable help

CASA design and implementation

A programmable command line interface with scripting
   (continued)

   In addition to toolkit: high-level tasks for the standard user

   tasks (implemented in Python) ⟶ **tools** (implemented in C++)

   e.g. the task *importfits* is based on the tool ***ia*** (image analysis):

```
#Python script
casalog.origin('importfits')
ia.fromfits(imagename,fitsimage,whichrep,whichhdu,zeroblanks,overwrite)
ia.close()
```

   CASA 5.1 comes with 132 implemented tasks.

   For ALMA analysis, the ALMA analysisUtils provide additional functionality

The ALMA analysisUtils

*The **ALMA analysisUtils** provide additional functionality*
for analysis and plotting. Often also useful for VLA and in general.

Originally developed for ALMA-internal Quality Assurance work.

Since a few years also public version available:

https://casaguides.nrao.edu/index.php/Analysis_Utilities

(provides Python module analysisUtils, typically renamed "aU")

CASA status and release plans
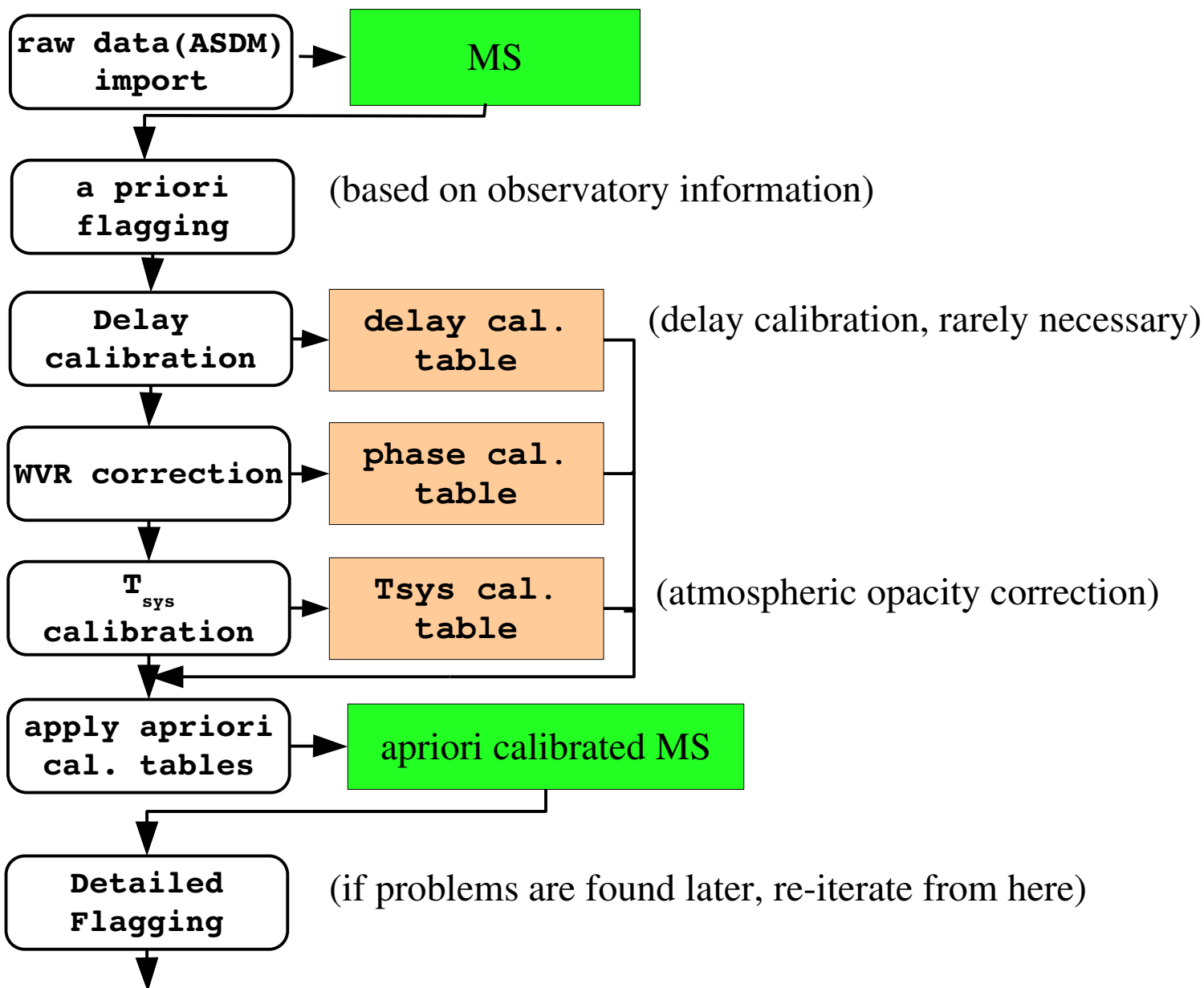
- Since Dec 2009 in public release under GPL = anybody can download,
  no warranty (see http://casa.nrao.edu),
  limited support (help desk, needs registration)

- Tutorials for the user community regularly given
  - Radio Interferometry: Methods and Science (Bonn, starting April 2018)
  - 16th Synthesis Imaging Workshop (NRAO, Socorro, NM) May 2018
  - ERIS school (next one 2019)
  - IRAM school (next one Oct 2018)

    look out for activities at your ALMA Regional Center Node

- The first public release was CASA 3.0.0 (Dec 2009), latest release 5.1.2 (January 2018),
  release 5.3 coming this Spring
  next ALMA-relevant release will be 5.4 near beginning of Cycle 6

- Development platforms: Linux (RHEL) + Mac OS X

- Supported platforms (binary distribution): RHEL 6 and 7, Mac OS X 11 and 12 (soon 13)

    - CASA normally works on other Linux flavours
    - on other Mac OSX versions this is more difficult

- Plans for SD development: support ALMA continuum (fast scan)
  and high-frequency (Band 9/10) observations

A typical ALMA interferometric analysis workflow with CASA

A typical ALMA interferometric analysis workflow with CASA

set model of flux cal. ← **flux cal. model**   (set model of flux calibration source)

phase-selfcal of bandpass cal → **phase cal. table**   (fit time-dep. of phase of bandpass cal.)

bandpass calibration → **bandpass cal. table**   (fit freq-dependence of gain of bandpass cal.)

gain calibration → **phase+amp cal. table**   (fit time-dependence of gain and phase of phase cal.)

scale gain cal. result → **phase+amp cal. table**   (scale the gain calibration result to the correct absolute flux)

apply gain+band calib. to target   (apply all calibrations)

calibrated MS

Where necessary, continuum subtraction and/or self-calibration will be applied.

A typical ALMA interferometric analysis workflow with CASA

```
┌─────────────────┐        ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
│   calibrated    │          Where necessary, continuum subtraction and/or
│       MS        │          self-calibration will be applied.
└─────────────────┘        └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘
         │
         ▼
┌─────────────────┐
│     imaging     │          (image the objects of interest using *tclean*)
└─────────────────┘
         │
         ▼
┌─────────────────┐
│  image (cube)   │
└─────────────────┘
      │      │                (analyse the image statistically etc., view
      ▼      ▼                 images and spectra and produce plots, export to
┌──────────┐ ┌──────────┐     FITS)
│numerical │ │ viewing, │
│analysis  │ │ plotting │
└──────────┘ └──────────┘
      │      │
      ▼      ▼
```

plots and numerical results

A typical ALMA SD analysis workflow with CASA

A typical ALMA SD analysis workflow with CASA

K to Jy/beam Conversion → **amp cal. table**

(conversion needed in prep. for combination, needs analysis of calibrator data, "Jy/K factors")

apply amp calibration

(using task *applycal*)

calibrated MS

With amplitudes in K in the DATA column and amplitudes in Jy/beam in the CORRECTED_DATA column

imaging

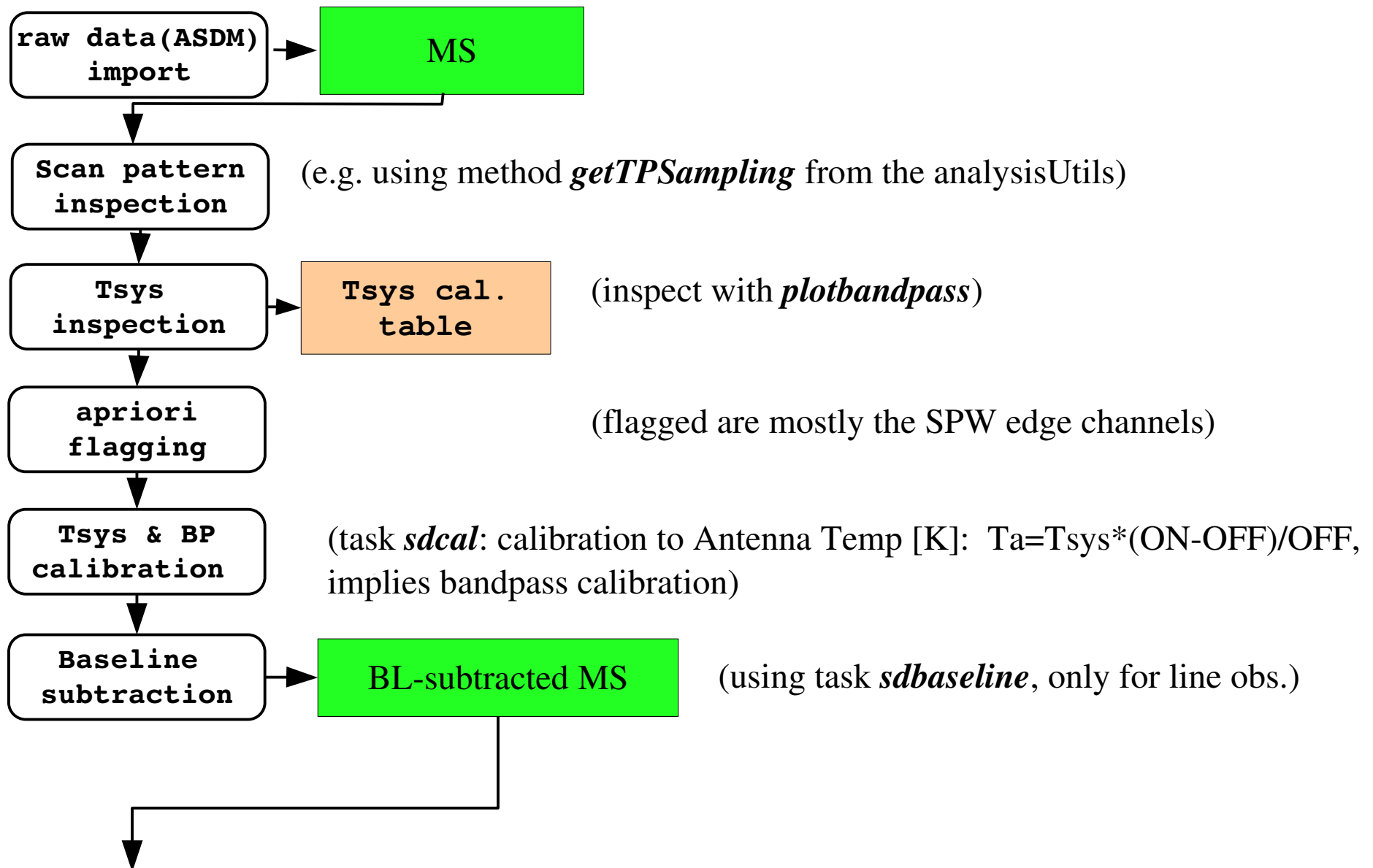(image the objects of interest using task *sdimaging*)

image (cube)

(analyse the image statistically etc., view images and spectra and produce plots, export to FITS)
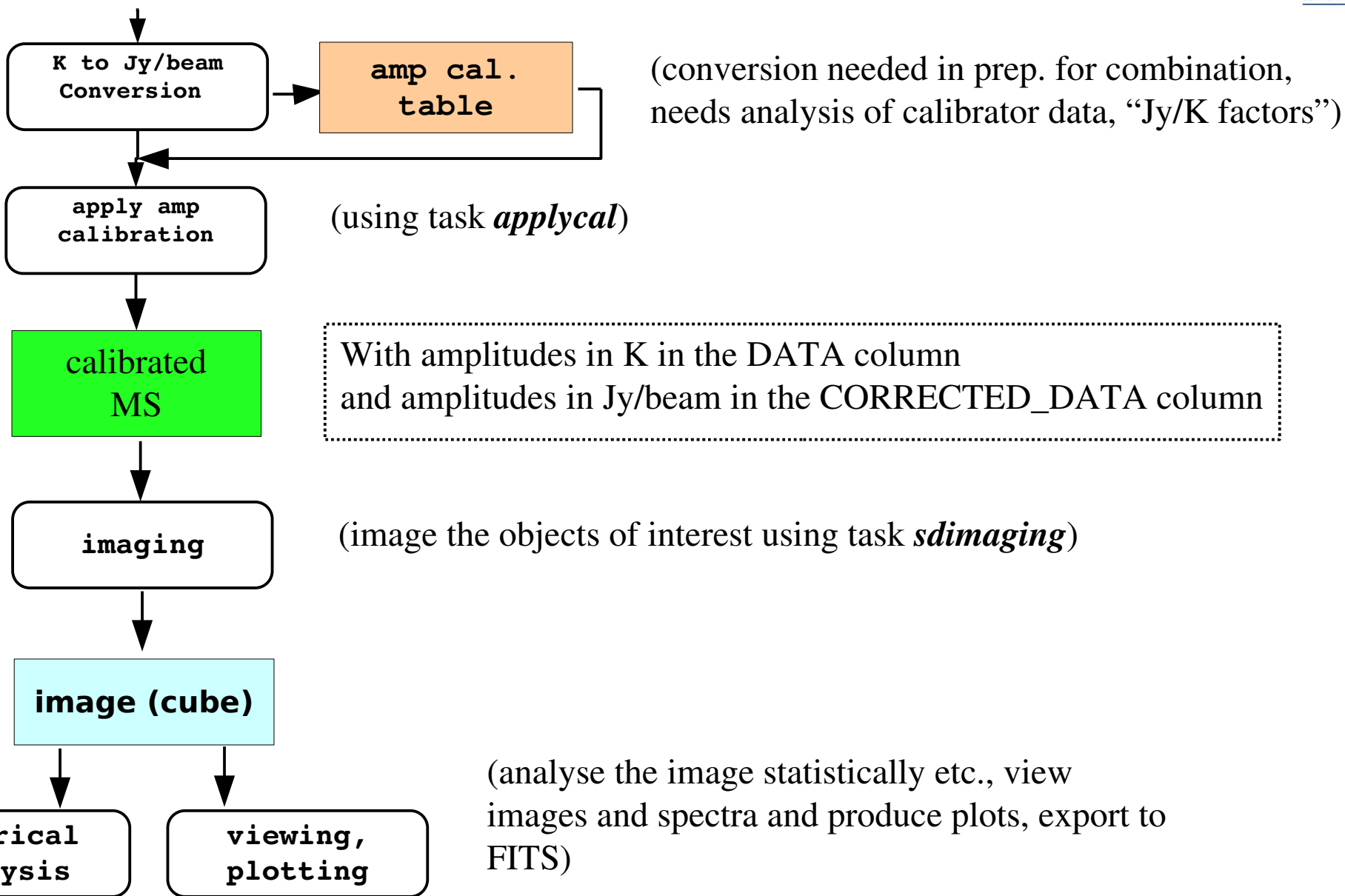
numerical analysis

viewing, plotting

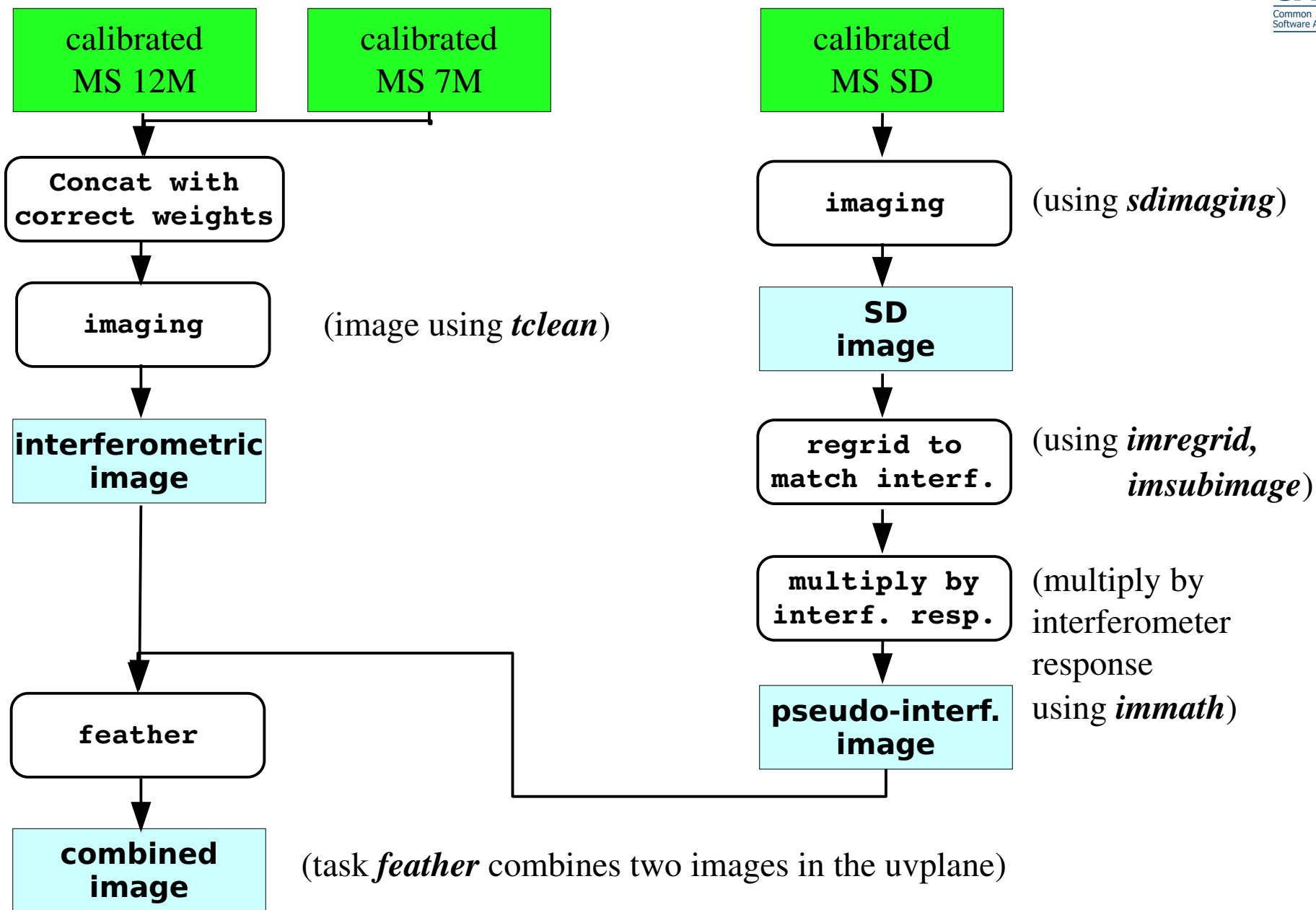# Combination of interf. and SD data workflow with CASA

calibrated MS 12M → Concat with correct weights → imaging (image using *tclean*) → interferometric image → feather → combined image

calibrated MS 7M → (into Concat with correct weights)

calibrated MS SD → imaging (using *sdimaging*) → SD image → regrid to match interf. (using *imregrid*, *imsubimage*) → multiply by interf. resp. (multiply by interferometer response using *immath*) → pseudo-interf. image → feather

(task *feather* combines two images in the uvplane)

Summary

- **The standard science data analysis package for ALMA and VLA is *CASA***
- **Data from other observatories can also be processed, e.g. BIMA, ATCA, SMA, ...**
- ***CASA* is mostly C++ code (libraries for general use available as *casacore*)**
- **> 20 FTEs are working on *CASA* + several more on the ALMA and VLA Pipeline in North America, Europe, Japan (+Australia and South Africa for casacore)**
- ***CASA* is a *comprehensive toolbox* with**
  - **MATLAB-like, scriptable user interface using *Python/IPython***
  - **procedures for calibration, imaging, spectral and spatial analysis, simulation and more**
  - **GUI tools for data selection, browsing, plotting, and image processing**
- **The command-line interface has two levels:**
  - ***tasks* for the common analysis problems**
  - ***tools* for everything else including *your own tasks***
- **the internal data format are *CASA Tables***
- **the *Measurement Set* is the *CASA* data format for visibility and SD data**
- **the interferometric calibration code is based on the *Measurement Equation***
- **single-dish analysis code development led by NAOJ, now fully MS-based**
- ***CASA* is publicly available under GPL for Linux and Mac OS X, installation is simple, see http://casa.nrao.edu/**
- **Two major releases per year, next one (5.3) will be out in Spring, ALMA uses 5.1.1**