CASA Pipeline for the CARMA/STING H I Data Reduction

Rui Xue[1]

1. Overview

...

2. Examples

...

3. Caveats

1. CONCAT() will merge spectral windows from different correlations into a single spectral window if they have the same channel setup. However, CLEAN() would expect a spectral window ID always has the same input polarization(s), so the second polarization setup of each spw it encounters will be ignored. In addition, if you have RR/LL dual polarization data and only RR or LL was flagged, CLEAN() will ignore both of them. To avoid merging spws with different correlation setups or throwing away all polarizations even just one goes bad, one can keep individual polarization under different spws by splitting and regridding them into slightly different channel setups.

2. Note that TOPO refers to a time stamp at a given observation date. If more than one obervation is concatenated this may lead to vastly erroneous values. Any conversion from TOPO to other frames such as BARY and LSRK should be performed for each individual observation, prior to concatenation or simultaneous imaging.

4. Reference

A. Files Structures

- ximport.py

  import/inspect/prepare visibility from uvfits/miriad/asdm/ms

- xcal.py

  calibrations: baseline correction / gain calibration / flux calibration

- xcalplot.py

  plotting calibration diagnostic figures

[1]Department of Astronomy, University of Illinois, Urbana-Champaign, IL 61801, USA

- xconsol.py

  extract calibrated source data / continuum subtraction / concatenate data from multiple tracks / adjust visibility weight

- xclean.py

  imaging/CLEANing spectral line / continuum data

- xutils.py

  functions supporting the pipeline

- init.py

  casapy initialization file for the pipeline setup

- xinit.py

  pipeline parameter initialization

- xexp.py

  experimental functions for testing

B. Functions in XUTILS

B.1. XUTILS.SCALEWT()

The theoretical variance of each visibility record can be calculated as follows:

$$\sigma_{ij}^2 = \frac{C_{ij}^2 T_i T_j}{2\Delta\nu\Delta T}, \tag{B1}$$

in which,

$$C_{ij} = \frac{2k_{\rm B}}{\sqrt{(\eta_i A_i)(\eta_j A_j)}} \frac{1}{\eta_q}. \tag{B2}$$

A and η are the antenna collecting area and aperture efficiency, respectively. η_q is the quantum efficiency from the backend (Koda et al. 2011). $\Delta\nu$ and ΔT are the channel width and integration time. To optimize the imaging quality, each visibility record will be assigned with a weight during the Fourier transform. In natural weighting, the weight is set to the inverse variance,

$$\mathrm{wt}_{ij} = \frac{1}{\sigma_{ij}^2}. \tag{B3}$$

For Briggs and uniform weighting[1], the weight are manipulated based on the basic visibility weight above and a gridding weight W_{ij} (proportional to the UV sampling density function).

A MIRIAD dataset includes the baseline-dependent variable JYPERK (different from JYPERKA), defined as

$$\mathrm{Jyperk} = \frac{2k_{\rm B}}{\sqrt{(\eta_i A_i)(\eta_j A_j)}}. \tag{B4}$$

[1] http://casa.nrao.edu/docs/casaref/imager.weight.html

Following Equation B3, the variance and visibility weight are calculated using the $T_{\rm sys}$ and Jyperk variables on-the-fly when they are requested by a task (e.g. INVERT, FITS, implemented in UVOI.C). However, η_q will be assumed to be 1. If a gain table exists, a modified Jyperk value will be used for the calculation (Jyperk$'$=$g_i g_j \times$Jyperk and g is the antenna-based gain amplitude here).

In CASA, the visibility weight is specified in the weight column of a measurement set (MS). The values are adjusted during calibrations with CALWT=TRUE using gain and $T_{\rm sys}$ tables when available, and the final weight in the calibrated MS are used as the basis for imaging. It essentially works as the same way how MIRAID/INVERT folds gains and $T_{\rm sys}$ into visibility weights with OPTIONS=SYSTEMP, but the current WEIGHT column implementation is handled correctly only in the relative sense:

- VLA archival data have the initial weight column when importing them into a MS, and the values are filled following Equation B1 using the bandwidth of each spectral window (spw). Defining the WEIGHT column as the bandwidth weight rather than channel weight is confusing for spectral line imaging, because the current spw regridding tasks (e.g. cvel(), mstransform()) don't modify the weight column when the channel width changes. The WEIGHT_SPECTRUM column (channel weight) is supposed to fill the blank, but it has not been implemented yet in most of calibration/imaging tasks.

- Raw EVLA data still have initial weight values of 1 for all records when filled into a MS. The new switch power gain and $T_{\rm sys}$ derived from MS SYSPOWER/CALDEVICE subtables (using GENCAL) can scale the weight values by,

$$\mathrm{wt} = \frac{1}{T_i T_j}. \tag{B5}$$

  But they are only available in recent EVLA data and still may have some drawbacks if not carefully handled. For example, bad $T_{\rm sys}$ and swpow gain readings might create odd calibrated visibility and weight values. In addition, scaling weights only using $T_{\rm sys}$ will not work when imaging a combined dataset with different backends / integrating time / channel width.

Although the above issues generally don't affect imaging individual observations, it might cause troubles when imaging a heterogenous dataset (e.g. VLA+EVLA) because the weight from different arrays could have incompatible absolute standards. To optimize the visibility weight of a heterogeneous dataset from different systems and observation setups, rescaling the weight is neccsary in CASA.

XUTILS.SCALEWT() will evaluate the statistical results of the line-free data noise derived from STATWT(), and rescale both WEIGHT and SIGMA columns, with the SIGMA values redefined as the estimated RMS noise for single channel and WEIGHT=1/SIGMA^2. The scaling approach avoids the uncertainty of noise estimations for individual record when using STATWT(), by keeping the relative weight pattern in the original data. In addition, it doesn't depend on an accurate values of all coefficients required to calculate the absolute variance.

Figure 1 shows the results of XUTILS.SCALEWT() from three VLA/EVLA HI 21cm observations (one 1999 VLA B-config track in black, one 2000 VLA BC-config track in red, and one 2013 EVLA B-config observation in orange) towards NGC 772. Before rescaling, the weight values in the calibrated EVLA data (without applying the switch power table) is smaller by a factor of > 1000 compared with the VLA data, causing the EVLA data incorrectly down-weighted. Considering differences in channel widths and

integration intervals, the expected weight in the absolute sense should follow:

$$\mathrm{wt} = 1/\sigma^2 \propto \frac{\Delta\nu\Delta T}{(T_{\rm sys}/\eta)^2}, \tag{B6}$$

Assuming $T_{\rm sys}/\eta$ decrease from 75K to 60K from VLA to EVLA in the L-band (Ott et al. 2008), the expected weight ratio between these VLA and EVLA tracks will be ∼10, according to the observation details summarized in Table 1. The results from XUTILS.SCALEWT() roughly matches this expectation.

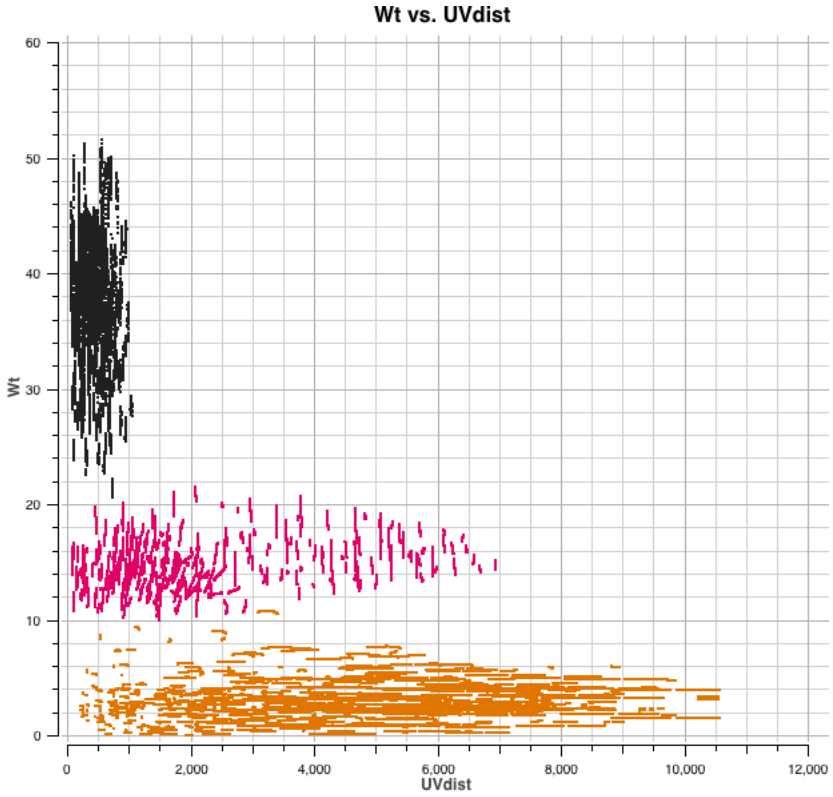

Fig. 1.— WEIGHT values vs. the uv-distance for three tracks of NGC 772. (this is from a dataset with rebinned spw, out-of-date)

B.2. XUTILS.IMPORTMIR() & XUTILS.IMPORTMIRIAD()

xutils.importmir() will fill the MIRAID data into a MS using the MIRIAD-uvfits-MS approach. It requires several MIRAID tasks, and is basically a Python wrapper for the whole data filling process.

Table 1. Weight Rescaling Results for NGC0772

Year	Array	Config	ChanWidth (KHz)	Pol.	Integ.Interval (s)	Weight (median)
1999	VLA	D	48.83	RR	60	13.7
2000	VLA	BC	97.66	RR	30	14.5
2013	EVLA	BC	62.50	RR LL	3	1.2

Alternatively, xutils.importmiriad() uses CASAFILLER (with several limitations and issues?).

B.3. XUTILS.XMOMENTS()

xutils.xmoments() is intended to replace the moments making algorithms we have in IDL[2]. Currently, only the 2D/3D-smooth+masking method is implemented.

B.4. XUTILS.RESMOOTHPSF()

Get the smoothed psf when using CLEAN() with option RESMOOTH=TRUE. In this case, the smoothed psf is the psf in the CLEANed image.

B.5. others

B.6. XUTILS.XPLOTCAL()

B.7. XUTILS.FLAGTSYS()

Flag bad $T_{\rm sys}$ the EVLA switch power table using median filtering and clipping.

B.8. XUTILS.COPYWEIGHT()

copying weight between the WEIGHT and WEIGHT_SPECTRUM columns

REFERENCES

Koda, J., Sawada, T., Wright, M. C. H., et al. 2011, The Astrophysical Journal Supplement

Ott, J., Perley, R., Rupen, M., & Team, E. 2008, in THE EVOLUTION OF GALAXIES THROUGH THE NEUTRAL HYDROGEN WINDOW. AIP Conference Proceedings

[2]`http://bitbucket.org/rxue/idl_moments`