CASA Pipeline for the CARMA/STING H I Data Reduction

Rui Xue[1]

1. Overview

...

2. Examples

...

3. Caveats

...

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

- 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

---

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

- 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 (from the MIRAID cookbook):

$$\sigma^2 = 1/weight = \frac{(Jy/K)^2 T_1 T_2}{2\Delta\nu\Delta T}, \tag{B1}$$

in which,

$$\frac{Jy}{K} = \frac{2k}{\eta A}, \tag{B2}$$

for the aperture efficiency η and aperture A. $\Delta\nu$ and ΔT are the channel width and integration interval respectively.

Before applying imaging weighting algorithms, MIRAID calculates the visibility weight using the $T_{\rm sys}$ and gain information on-the-fly during INVERTing, just following Equation B1 with modify Jy/K values. 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, just like the way how MIRAID folds gains and $T_{\rm sys}$ into visibility weights with OPTIONS=SYSTEMP. The final weight values in the calibrated MS are used as the basis for imaging weighting.

In both CASA and MIRAID, the weight is handled correctly in the relative sense for individual observation, although the absolute scale (defined by visibility noise) may be off by some factor. For example, VLA archival data does have initial weight values when importing them into a MS, and those values are presumably filled from observation metadata related to $T_{\rm sys}$. However, raw EVLA data (in MS format) usually have initial weight values of 1 for all records [1] Although this might not affect imaging individual tracks in the L-band ($T_{\rm sys}$ is relatively stable), it will cause troubles when combing VLA and EVLA data and imaging them together, because the weights from two different arrays have incompatible absolute standards.

XUTILS.SCALEWT() will evaluate the statistical results of the line-free data noise derived from STATWT(), and rescale both WEIGHT and SIGMA columns to the same absolute scale for individual observation by a single factor. In this way, the sigma and associated weight values are linked to meaningful physical units (noise of visibility) rather than an arbitrary standard of a specific dataset. At the same time, we could avoid the uncertainty of noise estimations for individual record when using STATWT(), by keeping the relative weight pattern in the original data.

---

[1]The new switch power gain and $T_{\rm sys}$ derived from MS SYSPOWER/CALDEVICE subtables (using GENCAL) can bring the weight values to meaningful physical units, 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.)

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:

$$weight = 1/\sigma^2 \propto \frac{\Delta\nu\Delta T}{(T_{\rm sys}/\eta)^2}, \tag{B3}$$

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

Although a reliable switch power calibration may eliminate the weight scaling in the future for a homogenous dataset, heterogeneous data from different systems (e.g. ALMA+CARMA) may still require weight adjustments before combing them for imaging.

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

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

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

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

B.4. others

B.5. XUTILS.XPLOTCAL()

B.6. XUTILS.FLAGTSYS()

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

B.7. XUTILS.COPYWEIGHT()

copying weight between the WEIGHT and WEIGHT_SPECTRUM columns

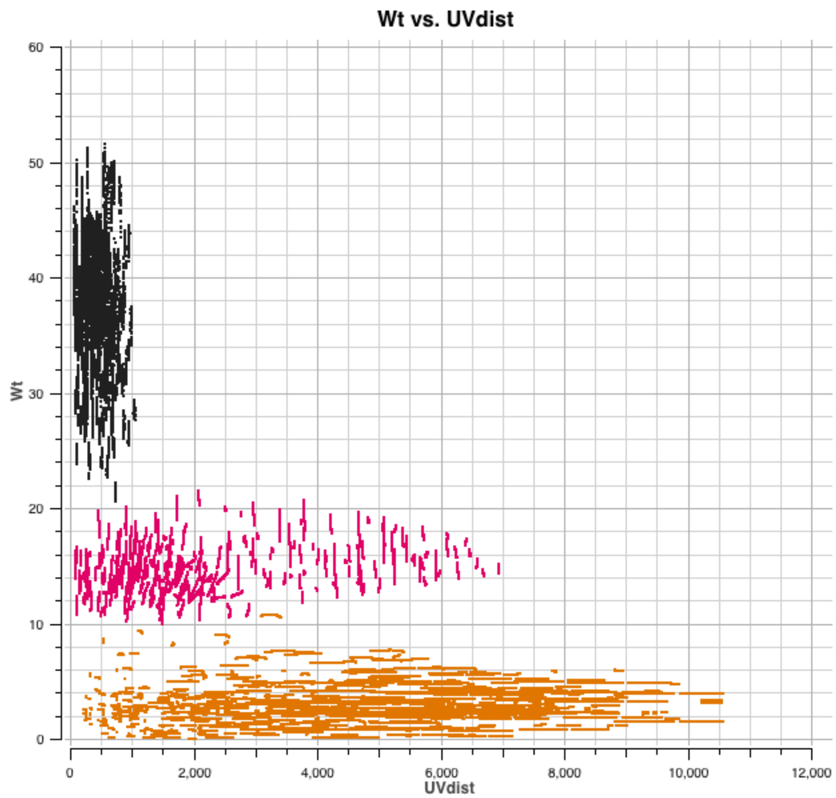

Fig. 1.— WEIGHT values vs. the uv-distance for three tracks of NGC 772. (this is from a dataset with rebinned spw, out-of-date)