

# I-TRAIN with the European ARC Network

## Fitting visibility data with UVMultiFit

S. Muller, Nordic ALMA Regional Center – January 2021

Visibility fitting can be a powerful method to analyze interferometric data and extract source information without the need to process image deconvolution, provided the source is “simple” enough. UVMultiFit [1] is a robust, versatile, and flexible Python-based tool dedicated to fitting visibility data, which can be called from within the CASA environment. This guide illustrates how to use UVMultiFit on real publicly-available interferometric data: we will fit point sources and extract spectra from data obtained with the ALMA array.

## 1 Requirements

To run the exercises, the user will need to have installed:

- the CASA package (the scripts were produced under CASA 5.6.2-3, other versions might work but without guarantee)
- the UVMultiFit software, available from the Github link:

`https://github.com/onsala-space-observatory/UVMultiFit`

- and the dataset and Python scripts which can be retrieved in the following tar file (1.8 Gb):

`https://bele.oso.chalmers.se/nordicarc/uvmultifit/Ex1/UVM\_Ex1.tar`

The tar file contains:

README\_UVM\_Ex1.pdf: Instructions guide

X2573.ave.ms: Calibrated measurement set of the ALMA observations

runUVM\_Ex1\_cont.py: the Python script for fitting two continuum point-sources

runUVM\_Ex1\_line.py: the Python script for extracting spectra toward point sources

## 2 The dataset

The data are taken from ALMA project 2013.1.00296.S [2]. They consist of band 7 ( $\sim 330$  GHz) observations of the quasar PKS 1830–211 obtained on 2015 May 19. To reduce data size, only one spectral window is used here (spectral window spw#1 in the original ALMA dataset), and the data were averaged in both time (time bin of 10 sec) and frequency (bin of two channels). The data were split with *mstransform*, moving the reference frame from TOPO to LSRK. Note that the data were pipeline-calibrated in a standard way, and no further self-calibration was applied.

The quasar PKS 1830–211 is lensed by an intervening  $z = 0.89$  galaxy into two bright and compact images, separated by 1 arcsec and embedded into an Einstein ring of 1 arcsec

in diameter. At the frequency and resolution of these ALMA data, only the two images are seen as point-like sources (see Fig. 1). Hence the dataset is well suited for visibility fitting analysis. The lines of sight to these lensed images intercept molecular clouds in the interstellar medium of the intervening galaxy, causing molecular absorption against the quasar continuum.

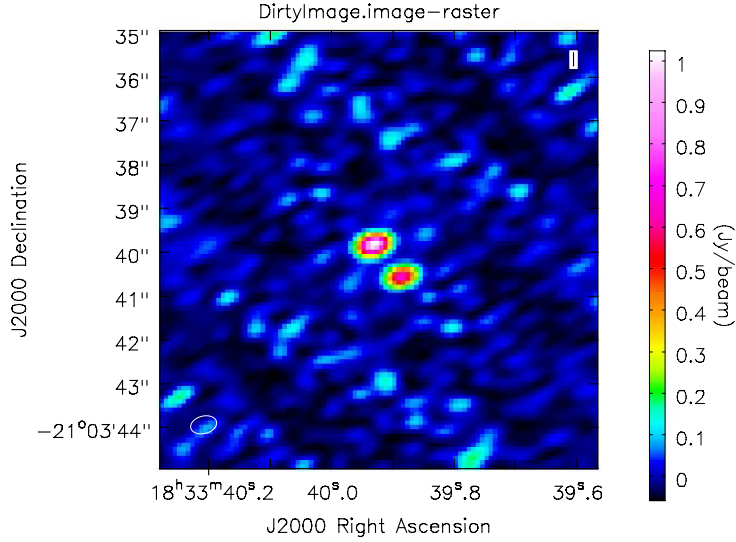


Figure 1: The lensed quasar PKS 1830–211: dirty (i.e., non-deconvoluted) map from the ALMA dataset showing the two lensed images separated by about 1 arcsec.

### 3 Fitting point sources

In a first step (*runUVM\_Ex1\_cont.py*), UVMultiFit is run on continuum channels to extract the positions of the two lensed images of the quasar, taken as point-sources (*model=['delta','delta']*). The first source is fitted by its position  $p[0], p[1]$  (offsets from the phase center, in arcseconds) and flux  $p[2]$ . The position of the second source is tied to that of the first one, i.e.,  $p[3]$  and  $p[4]$  are the offset (in arcsec) from the first source, and instead of its flux, we fit the flux ratio  $p[5]=\text{flux}_{\text{source1}}/\text{flux}_{\text{source2}}$ . The parameter *OneFitPerChannel=False* will make UVMultiFit to work on the continuum-collapsed cube (i.e., only one “continuum” channel).

Here is the corresponding call to UVMultiFit:

```
import uvmultifit as uvm
myuvfit = uvm.uvmultifit(vis='X2573.spw1.ms',
                        spw='0:300~800', # selection of continuum channels
                        field='PKS1830-211',
                        timewidth=1,
                        chanwidth=1,
                        model=['delta','delta'], # two point sources
                        var=['p[0],p[1],p[2]', 'p[0]-p[3],p[1]-p[4],p[2]/p[5]'],
```

```

p_ini=[0.0,0.0,1.0,0.642,0.728,1.5], # fit guesses
OneFitPerChannel=False, # to run on the continuum
column='data', # data column to be fitted
write='residuals',
outfile='UVMFitResults.cont.uvfit')

```

One can work with UVMultiFit results, which can be retrieved as:

```

fitval = myuvfit.result['Parameters']
fiterr = myuvfit.result['Uncertainties']

print "UVMULTIFIT RESULTS:"
for i in range(len(fitval)):
    print 'p[%d] = %.6f +/- %.6f'%(i, fitval[i], fiterr[i])

```

After the fit, one should find the following results:

```

p[0] = 0.170429 +/- 0.000038
p[1] = 0.065414 +/- 0.000027
p[2] = 1.066753 +/- 0.000160
p[3] = 0.646075 +/- 0.000070
p[4] = 0.723956 +/- 0.000049
p[5] = 1.545612 +/- 0.000425

```

The first source is located at a position (0.17,0.06) arcsec from the phase center (i.e., to the north-east) with a flux of about 1 Jy. The second source is offset by about (0.65,0.72) arcsec (i.e., to the south-west) with a flux 1.5 times lower than the first source.

The residuals after the fit are written on the 'corrected' column of the dataset (as set by the UVMultiFit parameter *write='residuals'*), although only for the subset of fitted channels (do not average visibilities on-the-fly during the fit if you want to output the residuals, i.e., make sure that both parameters *timewidth* and *chanwidth* are set to 1). To obtain an image of the fit residuals (e.g., with the *tclean* task, setting the parameter *niter=0*), also make sure to specify the same channel selection as for the UVMultiFit call.

For these ALMA observations of PKS 1830–211, the peak emission in the dirty image is  $\sim 1$  Jy and the rms noise level in the fit residuals is about 3 mJy, hence a dynamic range  $DR \sim 325$ . A round of self-calibration (not applied in this example) would definitely improve the quality of the data.

**Comment on the uncertainties from UVMultiFit:** The uncertainties are estimated from the Jacobian matrix, and scaled so that the reduced  $\chi^2$  equals unity. This assumes that the errors follow a Gaussian statistic and that there are no systematic errors introduced during the calibration. For the present dataset, the dynamic range  $DR$  is high and the data are clearly dynamic-range limited (see e.g., [3]). Therefore, the fit uncertainties listed above are likely under-estimated. As a rule of thumb for well-behaving data (e.g., well calibrated, not dynamic-range limited), one can expect a position accuracy  $\Delta$  of

$$\Delta \simeq \frac{\theta_{beam}}{2 \times SNR}, \quad (1)$$

where  $\theta_{beam}$  is the synthesized beam of the interferometric observations and  $SNR$ , the signal-to-noise ratio.

## 4 Fitting a spectral cube

In a second run (*runUVM\_Ex1\_line.py*), we want to extract the spectrum toward each lensed image of the quasar. The new UVMultiFit call looks like this:

```
pos1='0.170429,0.065413' # Position of the first image
pos2='-0.475656,-0.658543' # Position of the second image
myuvfitspec = uvm.uvmultifit(vis='X2573.spw1.ms',
                             spw='0:7~300', # channels around an H2O+ line
                             field='PKS1830-211',
                             timewidth=1,
                             chanwidth=1,
                             model=['delta','delta'],# two point-sources
                             var=[pos1+',p[0]',pos2+',p[1]'],
                             # fixed positions, only fit amplitudes
                             p_ini=[1.0,1.0],
                             OneFitPerChannel=True, # to get the spectrum
                             column='data',
                             outfile='UVMFitResults.spec.uvfit')
```

There, we have fixed the positions of the two point-sources (at *pos1* and *pos2*). Only their amplitudes will be fitted across channels (parameters  $p[0]$  and  $p[1]$ ) as set by the parameter *OneFitPerChannel*, which is now set to *True*.

In Fig. 2, one can see the resulting absorption spectra of the para- $\text{H}_2\text{O}^+$  (1,10–1,01 J=1/2–3/2) line (rest frequency of 634.266 GHz, redshifted to about 336 GHz for the  $z = 0.89$  absorber). Absorption is actually seen toward both images, with slightly different velocities (two different locations in the absorber).

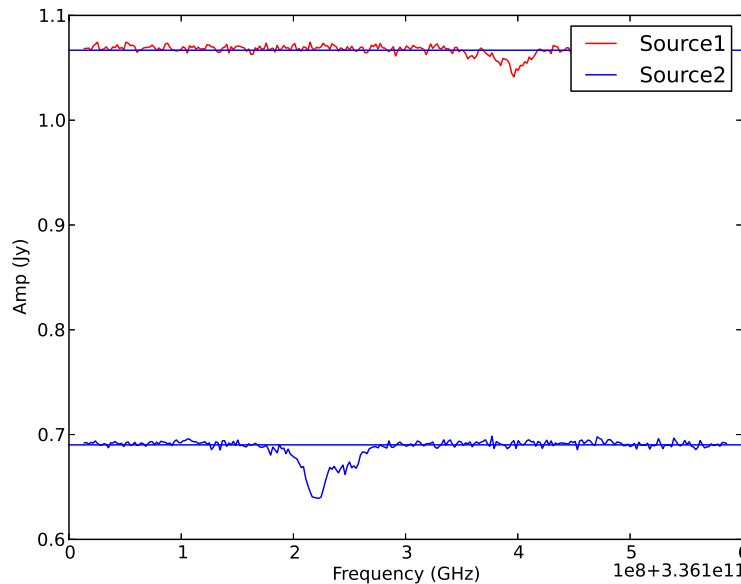


Figure 2: Absorption spectra of  $\text{H}_2\text{O}^+$  in the  $z=0.89$  absorber toward PKS 1830–211 [4], extracted with UVMultiFit. A model of two point-sources, whose positions were fixed at the locations of the two lensed images of the quasar, was fitted to the visibilities.

## Special trick

To avoid potential memory leakage in CASA, one can run multiple calls to UVMultiFit by opening external CASA sessions from within a main CASA (one needs "import os" before), like:

```
CASA> os.system("casapy_5.6.2-3 --nologger -c runUVM_Ex1_cont.py")
CASA> os.system("casapy_5.6.2-3 --nologger -c runUVM_Ex1_line.py")
```

Otherwise, in some circumstances, UVMultiFit can produce strange results after multiple calls to a given measurement set. In case you find strange results, you may also try to close your CASA session and restart a new run.

## References

- [1] Martí-Vidal, I., Vlemmings, W., Muller, S., & Casey S., 2014, A&A, 563, 136
- [2] Müller, H. S. P., Muller, S., Schilke, P., et al., 2015, A&A, 582, L4
- [3] ALMA Technical Handbook for Cycle 7, sec. 10.5.1
- [4] Muller, S., Müller, H. S. P., Black, J. H., et al., 2016, A&A, 595, 128