Data Comb 2019 for NGC346

Dirk Petry (ESO, Garching, Germany)
Lizette Guzman-Ramirez (Allegro, Leiden, Netherlands)
Daniel Tafoya (OSO, Onsala, Sweden)
Sandra Burkutean (Istituto di Radioastronomia, Italian ARC node, Bologna, Italy)
Alvaro Hacar (Allegro, Leiden, Netherlands)
Yusuke Miyamoto (NAOJ, Tokio, Japan)
George Bendo (UK ALMA Regional Centre Node, The University of Manchester, Manchester, UK)
September 4, 2019

This memo was prepared as part of the workshop "Improving Image Fidelity on Astronomical Data: Radio Interferometer and Single-Dish Data Combination," held on 12-16 Aug 2019 at the Lorentz Center in Leiden, The Netherlands.

1 Dataset overview

The dataset used by this group was extracted from the ALMA project 2017.A.00054.S (PI C. Agliozzo). It consists of a set of nine MOUSs of 7M data and nine MOUSs of TP data in Band 6. From these, the spectral range around the CO (2-1) line was extracted and averaged in order to make the dataset manageable in the short time of this workshop. The TP data was initially imaged by T. Stanke using initially a simple call to the sdimaging task. During the course of this workshop we realised that this left the image in a state with incorrect beam information and flux scaling. Then the image was re-created using the standard ALMA procedures which include a post-processing to adjust beam and scale.

Table 1: Observational details (edit as needed)

Parameter	Value
Phase center	00:59:05.089680 -72.10.33.24000 ICRS
Rest frequency	230.426 GHz
V_{LSR}	159.9 km/s
ΔV (channel width)	4.33 km/s
Velocity range imaged	30.3 km/s
Map size	$13' \times 13'$
12m-array pointings	none
7m-array pointings	1343
12m-array beamsize	$27''$
7m-array beamsize	$6.795 \times 6.340''$
Range of 12m baselines	n/a
Range of 7m baselines	[7.2 - 35.6] m

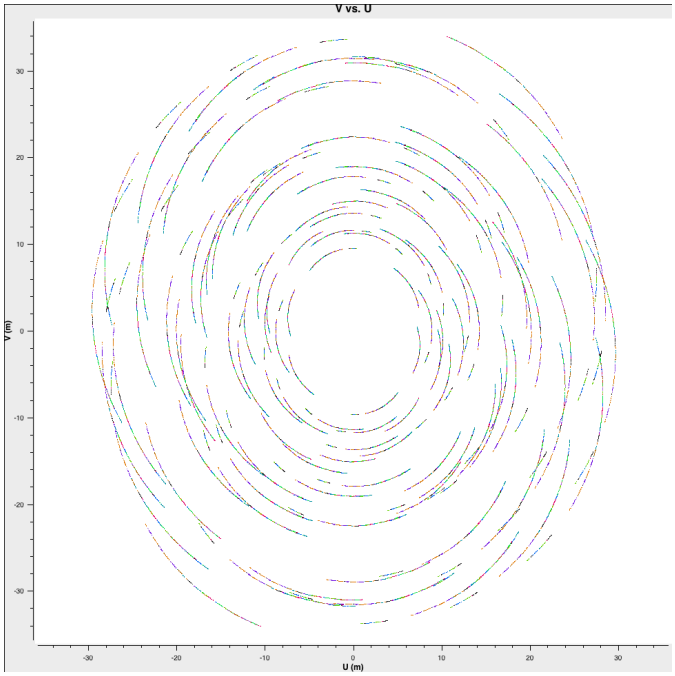


Figure 1: The uv coverage of the 7m data.

2 Combination Methods

2.1 Method 1: Feather

2.1.1 Combining a Cube

Problem #1: First error we found was that feather does not combine cubes if they do not have the same beam information format. One workaround was to split the cubes into individual channels.

```
Error: 2019-08-13 13:59:28 SEVERE feather::imager::feather() Caught exception:
    ↪ (../../images/Images/ImageBeamSet.cc : 109) Failed AlwaysAssert chan >=0 &&
    ↪ chan < Int(nchan()) && stokes >= 0 && stokes < Int(nstokes())
*** Error *** 2019-08-13 13:59:28 SEVERE feather::imager::feather() Caught
    ↪ exception: (../../images/Images/ImageBeamSet.cc : 109) Failed AlwaysAssert
    ↪ chan >=0 && chan < Int(nchan()) && stokes >= 0 && stokes < Int(nstokes())
2019-08-13 13:59:28 SEVERE feather:::: An error occurred running task feather.
```

The error above appears when trying to Feather spectral cubes from our TP and 7m data. The error is related to the different beam definitions in each of the two image headers (check header info with *imhead* command). In order to work with cubes, both input images need to have the same header parameters/definitions.

Solution: use *imsmooth(XXX, kernel='common')* command to unify header information:

```
imsmooth(imagename="NGC_346_sci.spw22.cube.I.manual-region3by3-7m-7bins-leiden.
    ↪ image.pbcor", kernel="common", outfile="TP_new_header")
```

Used in each of the two cubes, imsmooth creates one single beamsize for the entire cube. This solution can be used if the frequency range of the cube is small so the beam size/shape does not change significantly within the frequency range considered here.

Problem #2:

The other problem was that the SD beam size in the heather is 44.3879 arcsec, but it should be more like 27 arcsec. As indicated above, we later found that the initial SD image was lacking the standard ALMA postprocessing. In the weblog of the SD pipeline calibration, the beam is given as 32.6 arcsec. So there are 3 different beams. But this was reconciled later in the newly created SD image with postprocessing. The new SD image now has the expected beam of 27.96 arcsec.

2.1.2 How to estimate the sdfactor

How do we find the best sdfactor?

sdfactor = (SDImage -Feathered image_convolved to the SD beam)/SDimage

When trying to find the sdfactor, there's an error concerning the sizes of the images so the image needs to be re-sampled (using one as a template) . Although re-sampling the image changed the coordinates to ICS, and then the mathematical formula cannot be applied anymore.

Could it be that the SD is in Kelvins and the 7m in Jansky? Or the Jy/K conversion was done wrongly in the SD data? In the weblog review it says that the Jy/K conversion is a factor of 40. (Also this was later reconciled when the standard ALMA postprocessing was applied to the SD image.)

The CASA feather task says that "Plotting the data as a scatter plot is a useful diagnostic tool for checking for differences in flux scaling between the high and low resolution data sets. The dirty interferometer image contains the actual flux measurements made by the telescope. Therefore, if the single dish scaling is correct, the flux in the dirty image convolved with the low resolution beam and with the appropriate weighting applied should be the same as the flux of the low-resolution data convolved with the high resolution beam once weighted and scaled. If not, the sdfactor parameter can be adjusted until they are the same." This was not entirely clear. The latest version of CASA produced errors when reading in the high-resolution image (both in .fits and .im format). The axis labels were too small and the interface was not intuitive.

[**quote the actual CASA error message here**]

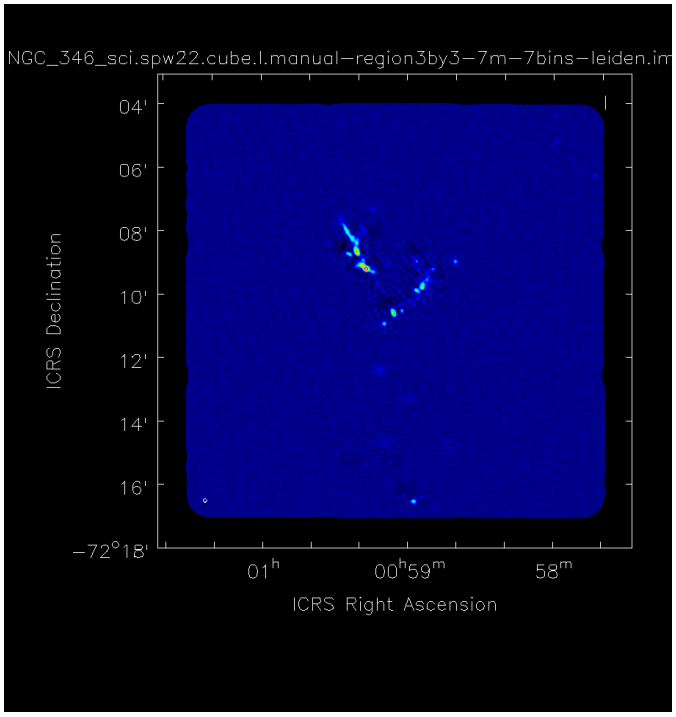
2.1.3 Notes on varying sdfactor

To test the effects of sdfactor on the image, a series of channel images were created varying sdfactor from 1 to 6. This clearly changed the relative weight of the single-dish data to the 7m interferometric data. Examples of this are illustrated in Figure 2.

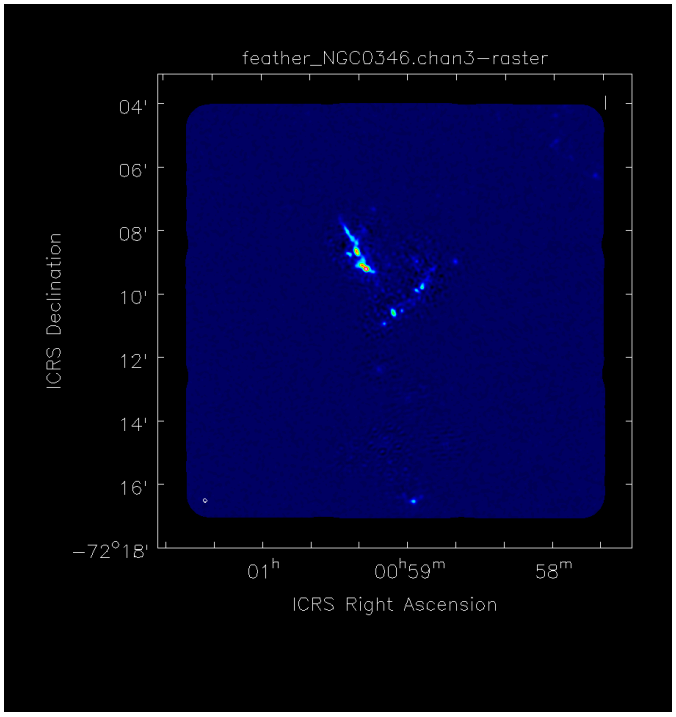
From a qualitative standpoint, an sdfactor of 1 leaves many of the hole-like artifacts in the data that are seen in the imaged based only on the 7m data. These artifacts disappear when sdfactor is set to 2 and continue to be less visible with higher sdfactor settings. However, even when sdfactor is set to 1, the peak value for the background shifts to above 0, which indicates that feathering has added back at least some extended emission, as can be seen in the histograms in Figure 3. Additionally, when the sdfactor is increased, it is clear that sources in the images become larger, even though the beam size attached to the image metadata does not vary. For example, the bright source at the bottom of channel 3 has a FWHM of 11.9×8.9 arcsec when sdfactor=1 but 29.1×23.5 arcsec when sdfactor=5.

These various issues make it difficult to judge what value of sdfactor should be used when applying feather to data.

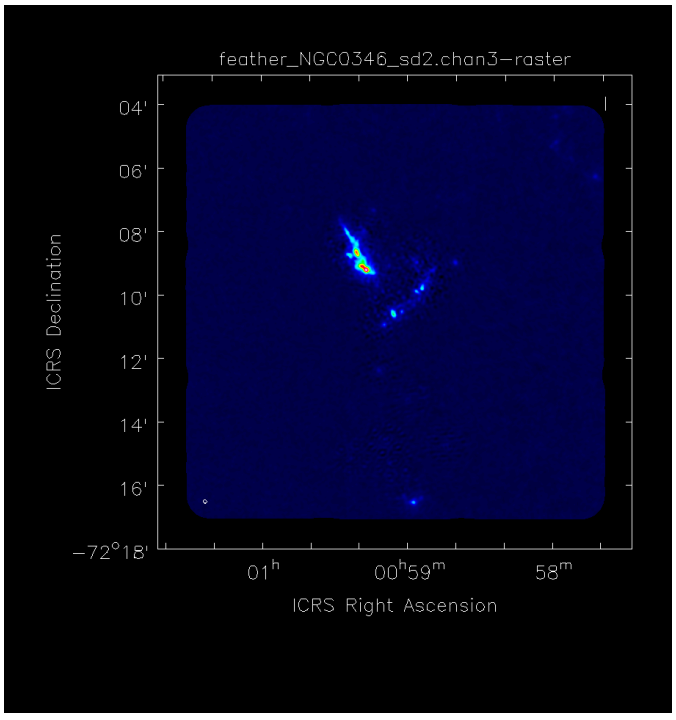
[**Make clear if the above results were obtained with the initial SD image or the one including postprocessing**]



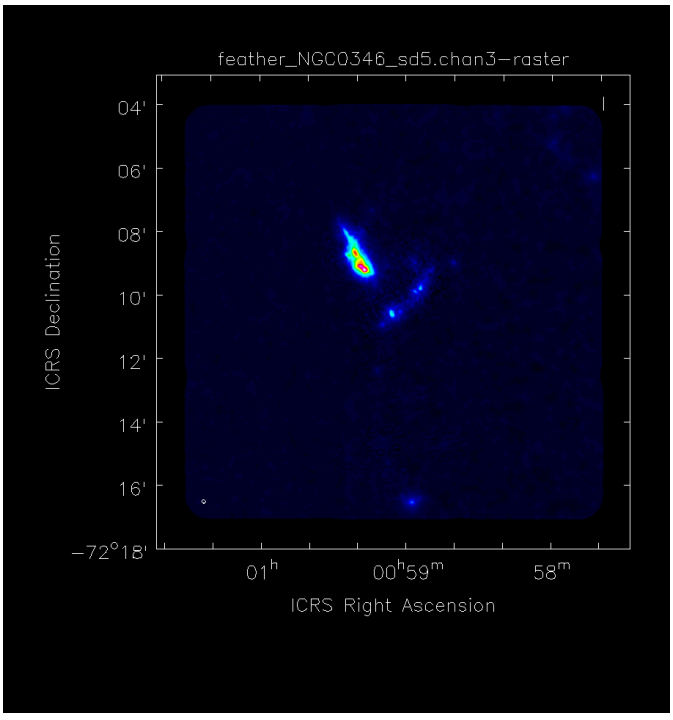
(a) The channel 3 7m data imaged by itself.



(b) The feathered version of the channel 3 data with no value set for sdfactor.

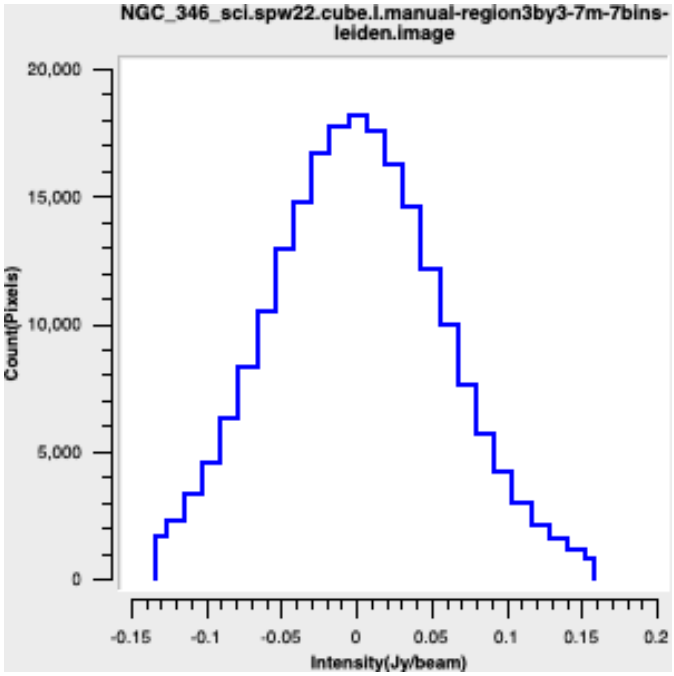


(c) The feathered version of the channel 3 data with sdfactor=2.

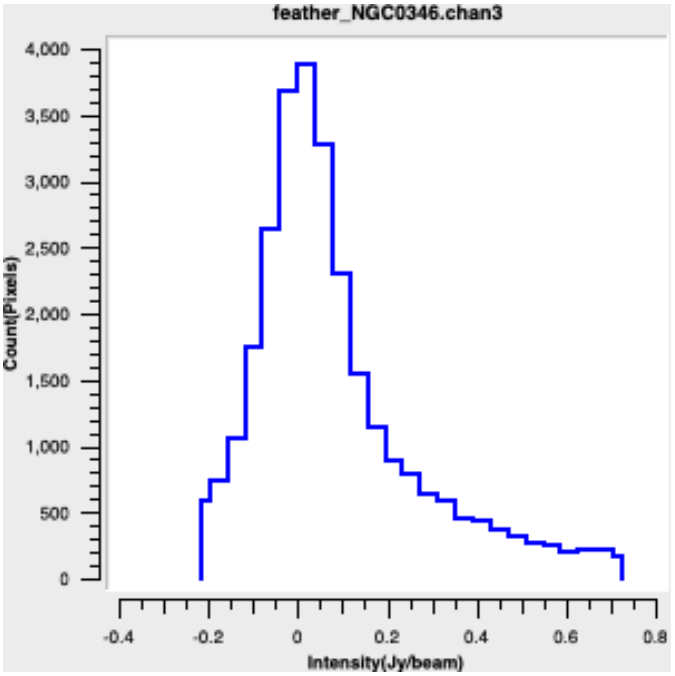


(d) The feathered version of the channel 3 data with sdfactor=5.

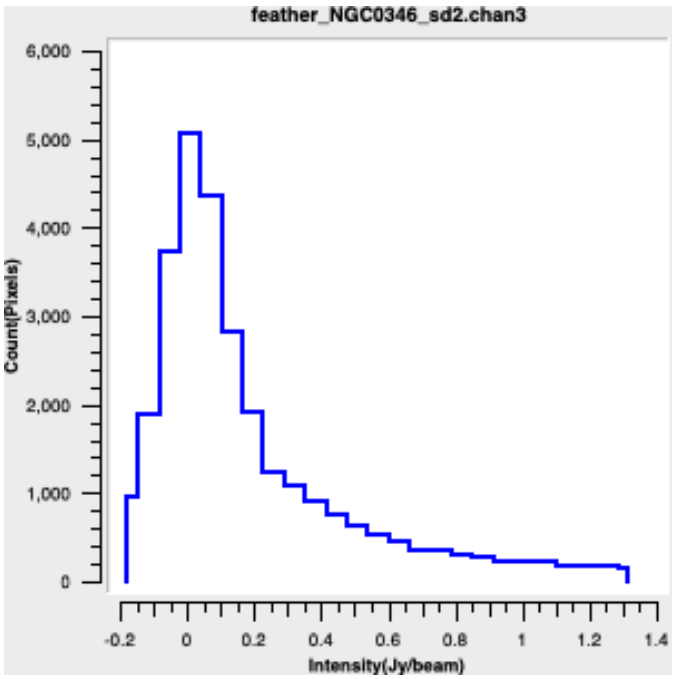
Figure 2: The image of the 7m data by itself for channel 3 as well as three versions of combining the single dish and 7m data with feather.



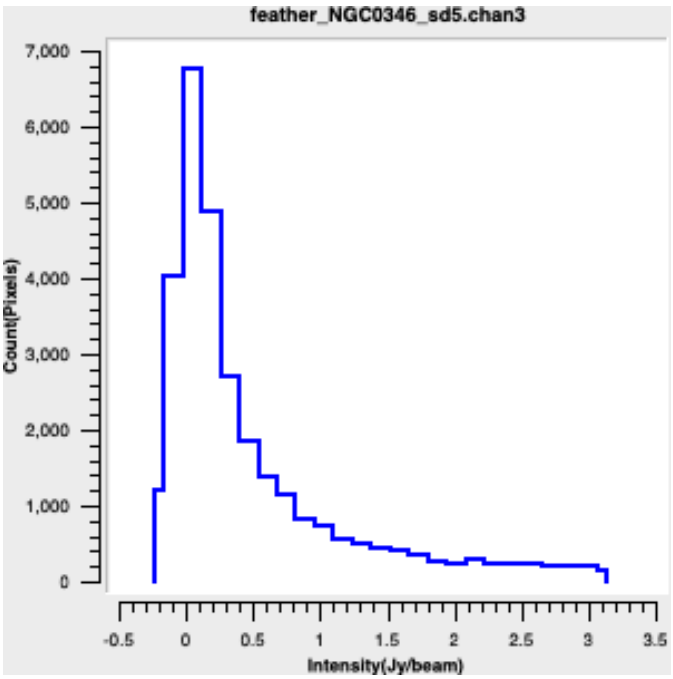
(a) The channel 3 7m data imaged by itself.



(b) The feathered version of the channel 3 data with no value set for sdfactor.



(c) The feathered version of the channel 3 data with sdfactor=2.



(d) The feathered version of the channel 3 data with sdfactor=5.

Figure 3: Histograms of the data from the bright sources near the centre of the four images shown in Figure 2. The histograms only show the central 98% of the data so as to show the distribution of the background values. High pixel valeus from the sources themselves are excluded.

2.2 Method 2: Joint Deconvolution (tp2vis)

2.2.1 TP2VIS Installation

The installation followed (on Linux, DP) largely the instructions provided on github:

```
cd tp2vis
git clone https://github.com/teuben/QAC.git
cd QAC
export PATH=/home/dpetry/temp/casa-from-tarball/casa-release-5.6.0-60.el7/bin:$PATH
# the above uses the existing CASA 5.6 installation rather than installing a new one
make data
make install
# install and run the benchmark
cd test
ln -s ../data/M100_TP_CO_cube.bl.image
ln -s ../data/M100_aver_7.ms
make bench
```

TP2VIS could then be initialised by starting CASA and typing

```
execfile('<path to QAC>/contrib/tp2vis.py')
```

and then started using lines like

```
tp2vis(infile='new.cube.spw23-forLeiden.image', outfile='new.cube.spw23-forLeiden.TP.m
```

2.2.2 Creating a pointing table for TP2VIS

Yusuke created the necessary pointing table using the ALMA OT based on the given SD image and using the necessary 12M spacing resulting in 4500 pointings.

Here are the steps on how to do this:

- set the source coordinate
- set the field size to match the 7m image mosaic extent
- download the pointing table (name in this case Alma12m.pointings)

Then do

Listing 1: get the relevant pointing info

```
cat Alma12m.pointings | awk "NR>2" | awk -F"," '{print $1,$2}' | sed 's/\([0-9]*\)
    ↪ \:\([0-9]*\):\([0-9.]*\) /\1h\2m\3 /' | sed 's/\([0-9][0-9]\)
    ↪ \:\([0-9][0-9]\)\:\([0-9][0-9]\)\./\1d\2m\3\./' | awk '{printf("ICRS␣\%␣ss␣
    ↪ \%␣ss\n",$1,$2)}' > 12m.ptg
```

2.2.3 Determining the "rms" parameter of TP2VIS

2.2.4 Running TP2VIS

TP2VIS was launched with various parameters:

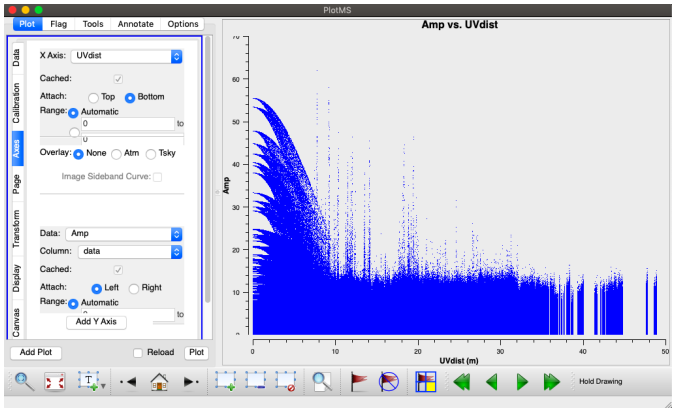
```
tp2vis(infile='new.cube.spw23-forLeiden.image', outfile='new.cube.spw23-
    ↪ forLeiden.TP.ms', rms=0.6, ptg='12m.ptg')
```

this completed successfully after ca. 30 minutes on a modern laptop and created an MS of 3.9 GB.

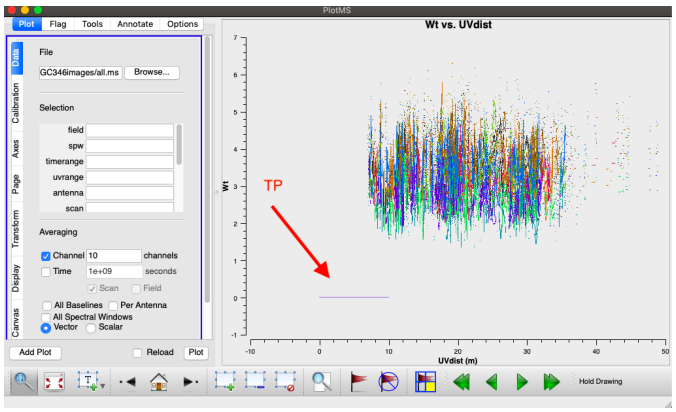
First issue found was that **the fields in this MS had all different names** although in a standard mosaic, all fields should all have the same name. This was communicated to Peter Teuben.

2.2.5 Cleaning the TP2VIS MS on its own

While analyzing the visibilities and weights of the two data sets we found the following. The visibilities of the combined data set seem very reasonable, on the other hand the weight of the TP data set is 4 orders of magnitude lower than the 7m data set (see figure **??**).



(a) Visibilities of the combined data sets showing combined visibilities.



(b) Weights of the combined data sets showing that the TP visibilities created with tp2vis are four orders of magnitude lower than the interferometric data set.

2.2.6 Joint cleaning of the TP2VIS MS and 7M MS

In attempts to clean on different laptops, different errors were observed.

One version of joint cleaning was attempted by using two different measurement sets as input to tclean. The tv2vis output for the channel 3 data was used as one of the measurement sets, and a subset of channels from the 7m visibility data were used for the second measurement set. The commands used are shown below.

```
mstransform(vis='region3by3-7M-leiden-avg_NGC_346.ms',
  outputvis='region3by3-7M-leiden-avg_NGC_346_chan3.ms',
  spw='0:228~235',
  datacolumn='data',
  chanaverage=True,
  chanbin=8,
  keepflags=False)
split(vis='new.cube.spw23-forLeiden.image.ms',
  outputvis='new.cube.spw23-forLeiden.image.chan3.ms',
```

```
  spw='0:3',
  datacolumn='data',
  keepflags=False)
os.system('rm␣-rf␣ngc0346_tp2vis_chan3.*')
tclean(vis=['new.cube.spw23-forLeiden.image.chan3.ms','region3by3-7M-leiden-
   ↪ avg_NGC_346_chan3.ms'],
  imagename='ngc0346_tp2vis_chan3',
  field='',
  gridder='mosaic',
  imsize=640,
  stokes = 'I',
  cell='1.4arcsec',
  phasecenter='ICRS␣00:59:05.089680␣-72.10.33.24000',
  weighting='natural',
  threshold='0mJy',
  specmode='mfs',
  outframe='LSRK',
  spw='',
  deconvolver = 'hogbom',
  niter=100,
  pbcor=True,
  interactive=False)
```

This produced the error below.

```
2019-08-14 14:07:24 SEVERE tclean::task_tclean:: Exception from task_tclean :
   ↪ Error in making PSF : Programmer error: called FFT2D with wrong size
2019-08-14 14:07:24 SEVERE tclean:::: An error occurred running task tclean.
```

While running tclean using the concatenated visibilities experienced several issues: Here are the commands used:

```
tp2vis('new.cube.spw23-forLeiden.image','new.cube.spw23-forLeiden.image.ms','12m.
   ↪ ptg',rms=0.5) \\
concat(vis=['region3by3-7M-leiden-avg.ms','new.cube.spw23-forLeiden.image.ms'],
   ↪ concatvis='all.ms',copypointing=False)
```

1. Using the full tclean commands

```
tclean(vis = 'all.ms',
        imagename = 'all_tp2vis.image',
        field = '0~4499',
        intent = 'OBSERVE_TARGET#ON_SOURCE',
        phasecenter = 3, #00:59:05.089680 -72.10.33.24000 ICRS
        stokes = 'I',
        spw = '0',
        outframe = 'LSRK',
        restfreq = '230.538GHz', # CO 2-1 lab rest freq
        specmode = 'cube',
        imsize = [640, 640], # 3500 arcsec x 2000 arcsec
        cell = '1.4arcsec',
```

```
        deconvolver = 'hogbom',
        niter = 100,
        weighting = 'briggs',
        robust = 0.5,
        mask = '',
        gridder = 'mosaic',
        pbcor = True,
        threshold = '0.02Jy',
        width = '3.333MHz',
        start = '230.405GHz',
        nchan = 7,
        interactive = True
        )

2019-08-15 07:42:57 SEVERE tclean::task_tclean:: Exception from task_tclean :
   ↪ Error in making PSF : (../../casa/OS/Path.cc : 420) Failed AlwaysAssert
   ↪ getcwd(temp, 1024)
libc++abi.dylib: terminating with uncaught exception of type casacore::AipsError:
   ↪ (../../casa/OS/Path.cc : 420) Failed AlwaysAssert getcwd(temp, 1024)
tar: 4D4A0FD0-FD59-45EB-8AA4-5802A8E752FB.dmp: Cannot stat: No such file or
   ↪ directory
tar: Error exit delayed from previous errors.
rm: 4D4A0FD0-FD59-45EB-8AA4-5802A8E752FB.dmp: No such file or directory

--------------------------------------------------------------------
CASA has crashed...
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
A crash report is being generated and submitted. The report
will contain information about CASAâĂŹs environment when the
crash occurred (e.g. CASA log, call stack, CPU information).
This should not take too long...
--------------------------------------------------------------------

Abort trap: 6
```

2. Using the tclean commands to image only one channel

```
tclean(vis = 'all.ms',
        imagename = 'all_tp2vis.image',
        field = '0~4499',
        intent = 'OBSERVE_TARGET#ON_SOURCE',
        phasecenter = 3, #'ICRS 00:59:05.08 -72.10.33.24'
        stokes = 'I',
        spw = '0:3',
        outframe = 'LSRK',
        restfreq = '230.538GHz', # CO 2-1 lab rest freq
        specmode = 'mfs',
        imsize = [640, 640], # 3500 arcsec x 2000 arcsec
        cell = '1.4arcsec',
```

```
        deconvolver = 'hogbom',
        niter = 100,
        weighting = 'briggs',
        robust = 0.5,
        mask = '',
        gridder = 'mosaic',
        pbcor = True,
        threshold = '0.02Jy',
        width = '3.333MHz',
        start = '230.405GHz',
        interactive = True
        )
2019-08-15 08:12:18 SEVERE tclean::task_tclean:: Exception from task_tclean :
   ↪ Error in Weighting : (../../casa/OS/Path.cc : 420) Failed AlwaysAssert
   ↪ getcwd(temp, 1024)
2019-08-15 08:12:18 SEVERE tclean:::: An error occurred running task tclean.
```

3. Using the tclean commands to image only one channel and removing the weighting parameters

```
tclean(vis = 'all.ms',
        imagename = 'all_tp2vis.image',
        field = '0~4499',
        intent = 'OBSERVE_TARGET#ON_SOURCE',
        phasecenter = 3, #'ICRS 00:59:05.08 -72.10.33.24'
        stokes = 'I',
        spw = '0:3',
        outframe = 'LSRK',
        restfreq = '230.538GHz', # CO 2-1 lab rest freq
        specmode = 'mfs',
        imsize = [640, 640], # 3500 arcsec x 2000 arcsec
        cell = '1.4arcsec',
        deconvolver = 'hogbom',
        niter = 100,
        mask = '',
        gridder = 'mosaic',
        pbcor = True,
        threshold = '0.02Jy',
        width = '3.333MHz',
        start = '230.405GHz',
        interactive = True
        )

2019-08-15 08:18:01 SEVERE tclean::task_tclean:: Exception from task_tclean :
   ↪ Error in making PSF : Cannot open existing image : all_tp2vis.image.psf :
   ↪ Error in opening Image : all_tp2vis.image.psf
2019-08-15 08:18:01 SEVERE tclean:::: An error occurred running task tclean.
```

2.2.7 Re-imaging tp2vis output by itself

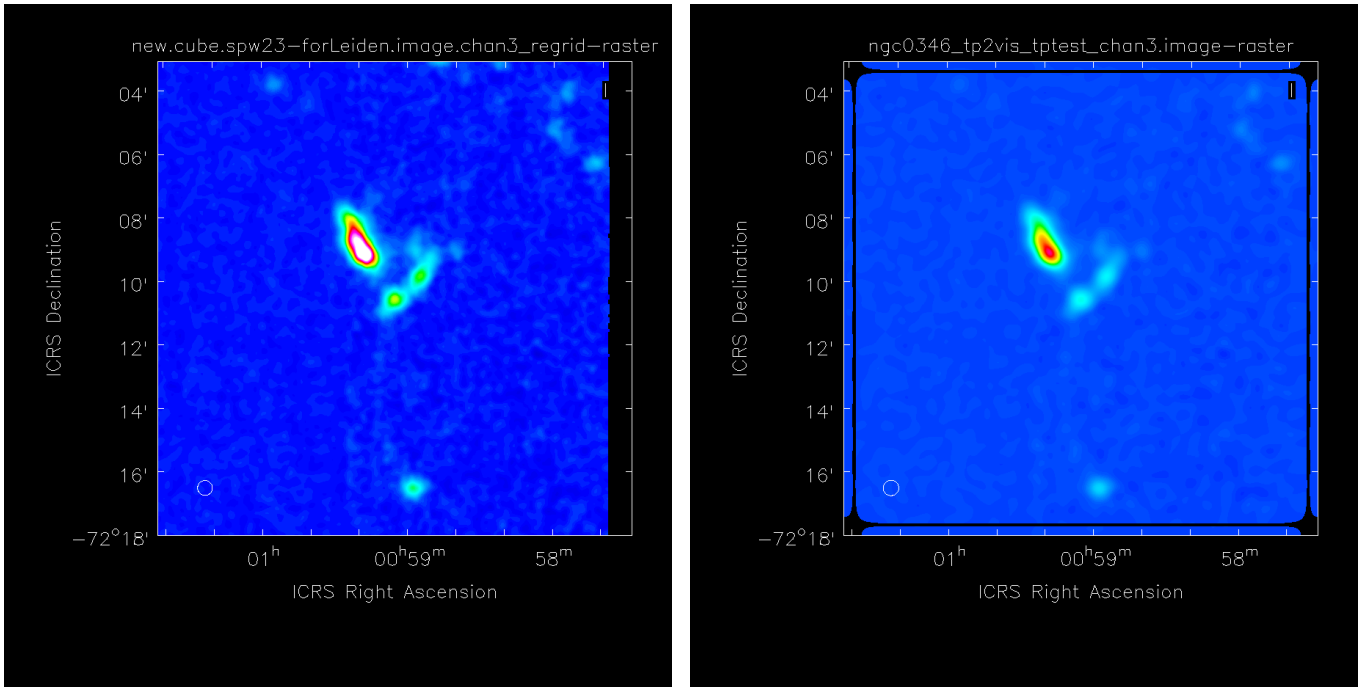
Given the technical difficulties in imaging both the 7m visibility data and the tp2vis output as a single image or image cube, we tested whether it was possible to image just the output from tp2vis. We used the following commands:

```
split(vis='new.cube.spw23-forLeiden.image.ms',
  outputvis='new.cube.spw23-forLeiden.image.chan3.ms',
  spw='0:3',
  datacolumn='data',
  keepflags=False)
tclean(vis='new.cube.spw23-forLeiden.image.chan3.ms',
  imagename='ngc0346_tp2vis_tptest_chan3',
  field='',
  spw='0',
  stokes = 'I',
  imsize=640,
  cell='1.4arcsec',
  phasecenter='ICRS 00:59:05.089680 -72.10.33.24000',
  specmode='mfs',
  outframe='LSRK',
  weighting='natural',
  deconvolver = 'hogbom',
  niter=100,
  threshold='0mJy',
  gridder='mosaic',
  pbcor=True,
  interactive=True)
```

The output is shown in Figure 1 alongside a version of the original image regridded to the same coordinate system. The images look largely similar in appearance. However, the images are different in a few ways. Most importantly, the flux density for sources in the images is $sim3\times$ lower than the corresponding flux densities in the original images. The rms noise is lower by $\sim 4.5\times$ in the tp2vis output, which indicates that, even when the flux scaling effects are taken into account, the data are smoothed by this process. The beam area in the tp2vis output is also 10% larger.

As an additional weird note, the PSF image produced by this tclean command contains a reasonable-looking PSF image at its center but also contains noise outside the centre that has an rms of $\sim$1% of the peak. See Figure 6.

Figure 7 shows the results with some weights. Another additional weird point is that primary beam depends on the weight.



(a) Original total power image in channel 3.

(b) Image created from tp2vis output for channel 3 using tclean.

Figure 5: Comparison of original total power image to image created by re-imaging tp2vis output.

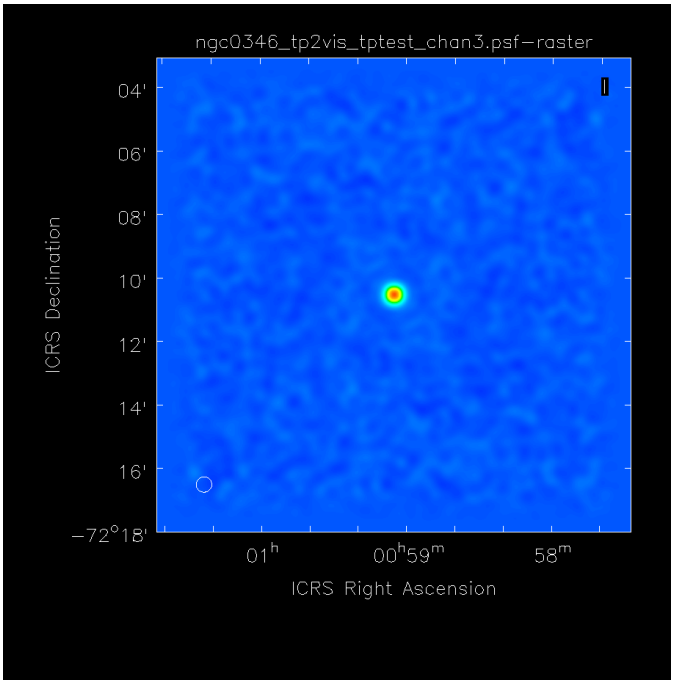


Figure 6: PSF created when re-imaging tp2vis output.

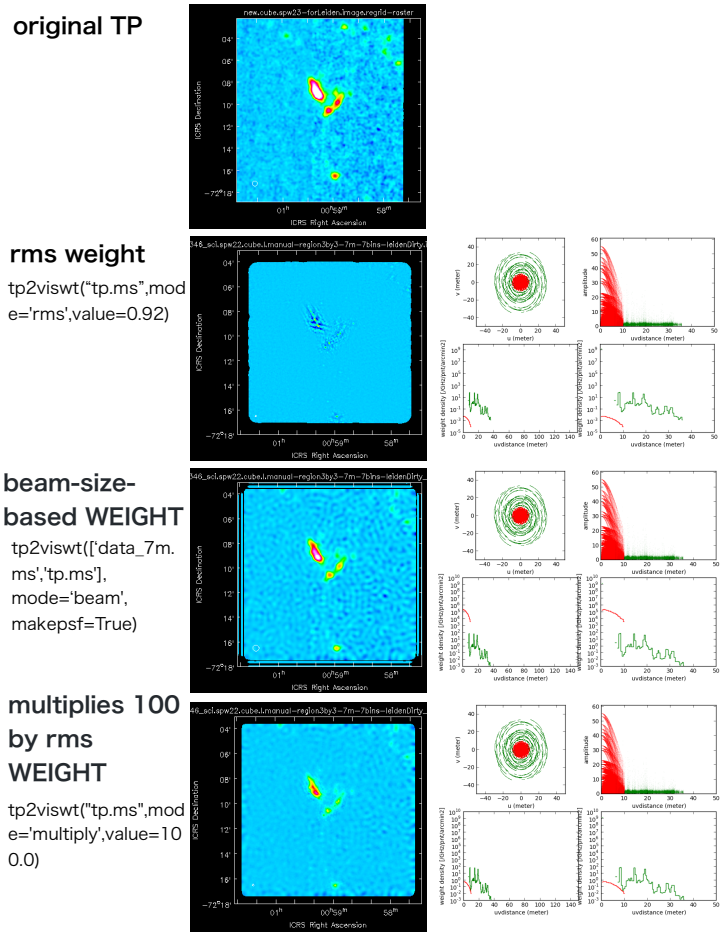


Figure 7: Effect of weight for combining TP and 7m.

2.3 Method 3: SDInt

As described to us by Tim Braun, sdint was installed and applied to our NGC 346 data.

The sdint installation consists of three Python files: sdint_imager.py, sdint_helper.py, and the script to actually perform the imaging steps named runsdint.py .

In order to get useful results, several modifications had to be applied to the code:

sdint_imager.py: the parameters "start" and "interactive" needed to be exposed in order to be able to specify our cube grid and perform interactive masking.

sdint_helper.py: no changes necessary.

runsdint.py: besides the obvious file name changes, the deconvolver "hogbom" and specmode "cube" was chosen. Other parameters which needed modifying were phasecenter, imsize, cell, reffreq, dishdia, start, width, nchan.

Furthermore, sdint needs an SD PSF image. In Tim Braun's example, this is obtained by taking an existing SD image, blanking it and then taking the per-channel-beam entries in the header and creating centered symmetric Gaussians in each channel of the cube (σ = major axis).

In our example, the existing SD image contained only a common beam. The solution was to take the major axis of that common beam and generate a pseudo-per-channel beam with all channels having the same beam.

Furthermore, experiments showed that **the center of the SD PSF image has to be identical with the phasecenter of the interferometric image to be created! Otherwise, severe image artifacts arise like the massive "hole" on the left side of the image shown in figure 8.**

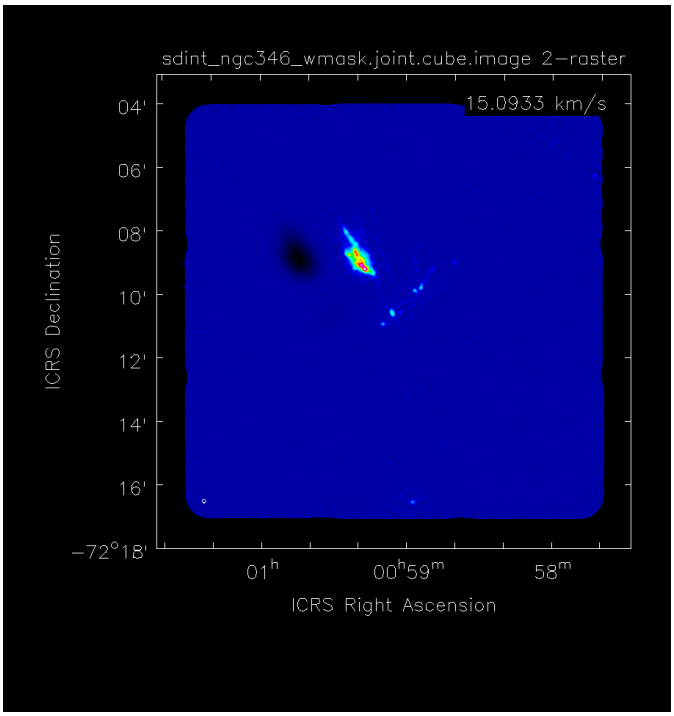


Figure 8: The channel 3 image created by sdint with the "hole" artifact present (caused by the SD PDF image not having the same center as the 7M mosaic).

With the "interactive" parameter exposed in sdint_imager.py, the convenient, standard interactive clean was possible. No user interface problems were encountered.

The final best image (again only channel 3 shown which is the one with the most bright and complex emission) is shown in figures 9 (standard scaling) and 10 (scaling changed to show noise properties).

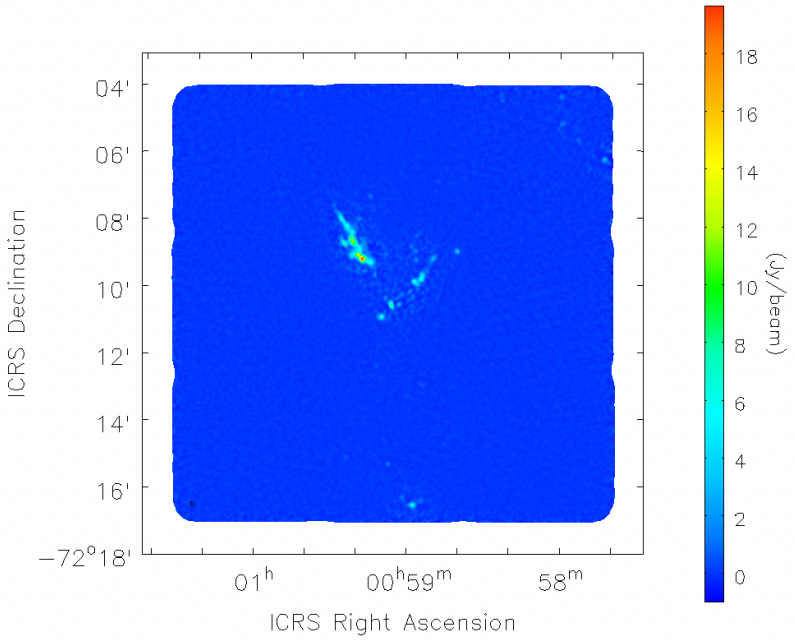


Figure 9: The final channel 3 image created by sdint using masking and an SD input PSF image centered on the interferometric phase center.

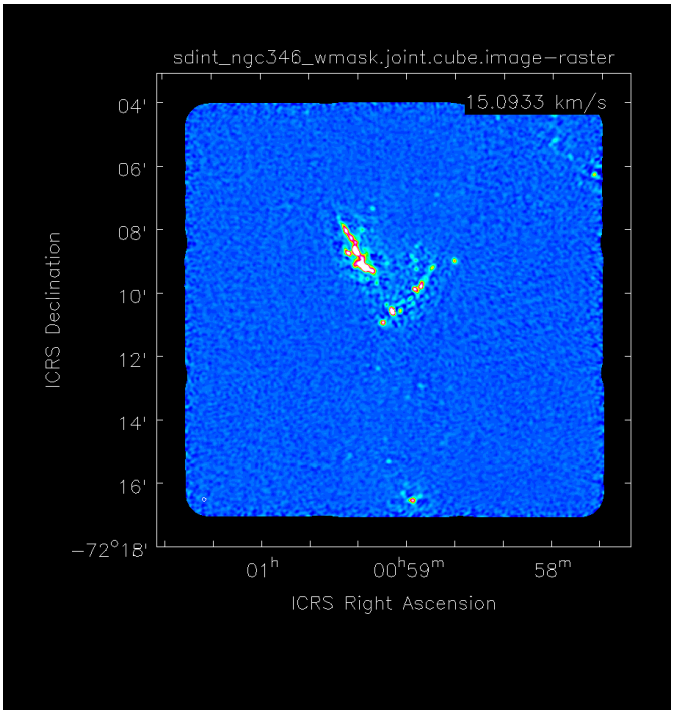


Figure 10: like figure 9 but with scaling emphasizing the noise properties.

3 Comparison and evaluation

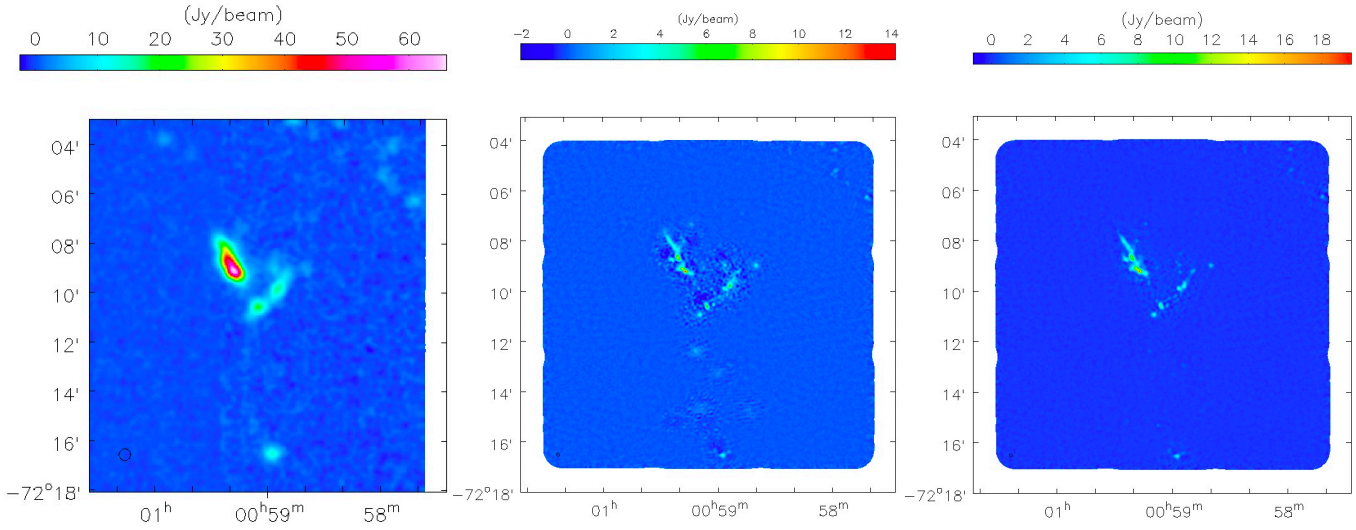


Figure 11: Comparison between the original TP image (left), 7m-interferometric image (center) and an image created with SDINT (right).

3.1 Image fidelity plots

The ratio of two images defines an initial test to quantify the differences between distinct data reductions:

$$Ratio_\nu(x,y) = \frac{Int1_\nu(x,y)}{Int2_\nu(x,y) - Output_\nu(x,y)} \tag{1}$$

where $Ratio_\nu(x,y)$ is the flux ratio at the (x,y) position and ν frequency (or channel) between images #1 $Int1_\nu(x,y)$ and #2 $Int2_\nu(x,y)$.

ALMA/CASA defines the Image Fidelity parameter as:

$$F_\nu(x,y) = \frac{Output_\nu(x,y)}{Input_\nu(x,y) - Output_\nu(x,y)} \tag{2}$$

where $F_\nu(x,y)$ is the fidelity at a given (x,y) position and ν frequency (or channel), and $Input_\nu(x,y)$ and $Output_\nu(x,y)$ are the emission in the input and output images at the same position and frequency, respectively [**give exact reference for this definition**]. In the case of simulations, the Input image refers to the true sky emission while the output image corresponds to the final image obtained after being observed by the interferometer and recovered via different methods (see above).

In observations, and in the absence of the true sky emission at the interferometric resolution, it is possible to assume the single-dish image (SD) as the true sky emission at the SD resolution (assuming no calibration issues). Then, the recovered interferometric image can be convolved into this final resolution to test the total recovered flux at this same resolution. In this case, the Image Fidelity at the SD resolution ($F_{\nu,SD}(x,y)$) can be estimated as:

$$F_{\nu,SD}(x,y) = \frac{Int_{\nu,SD}(x,y)}{SD_\nu(x,y) - Int_{\nu,SD}\nu(x,y)} \tag{3}$$

where $Int_{\nu,SD}(x,y)$ is the recovered emission by the interferometer at the position (x,y) and frequency ν at the resolution of the SD compared to the true SD emission $SD_{\nu,SD}(x,y)$ at the same position and frequency. In both cases, higher $F_\nu(x,y)$ and $F_{\nu,SD}(x,y)$ represent better agreement between the true sky emission and the recovered emission by the interferometer + combination methods.

Alternatively, one could also define the Goodness parameter as:

$$Q_\nu(x,y) = \frac{SD_\nu(x,y) - Int_{\nu,SD}(x,y)}{SD_\nu(x,y)} \tag{4}$$

where $Q_\nu(x,y)$ is the goodness parameter at a given position (x,y) and frequency (aka channel) ν, $SD_\nu(x,y)$ is the single-dish emission at the same given position, $Int_{\nu,SD}(x,y)$ the interferometric emission (after full combination) convolved into the SD resolution. Compared to the image Fidelity, the Goodness parameter is approximately $Q_\nu(x,y) \sim 1/F_\nu(x,y)$. Therefore, low $|Q_\nu(x,y)|$ values define a good agreement between the sky and interferometric images where $|Q_\nu(x,y)|$ defines the percentage (%) of this agreement in comparison to the SD image.

3.2 Difference image

Comparison of the feathered image with the SDINT final image. The output from SDINT has a header with per-channel beams, so one has to use imsmooth. The final difference image is seen in Figure 12. SDINT seems to recover more flux.

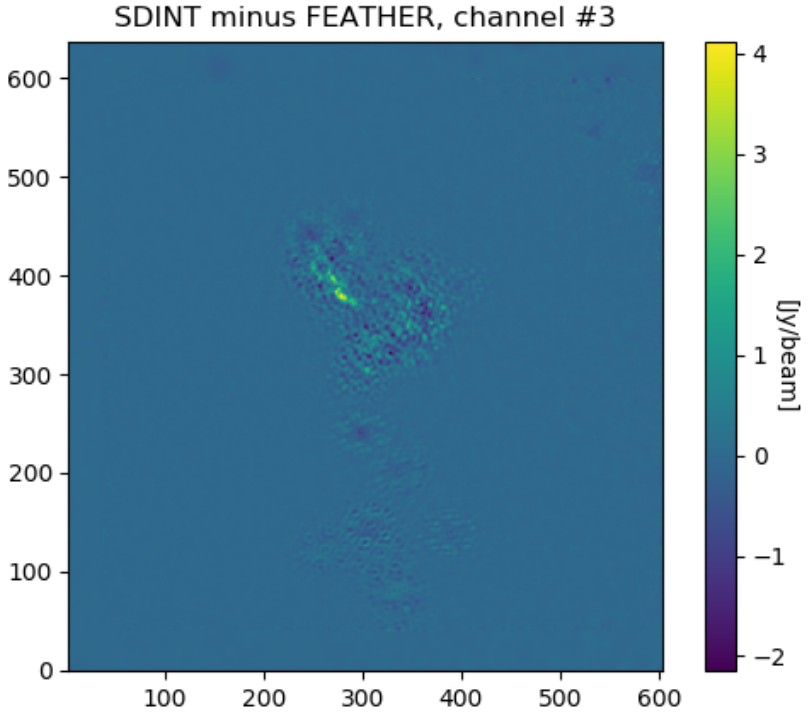


Figure 12: The difference image between the SDINT result and the FEATHERED image (sdfactor = 1.0) for channel number 3.

3.3 Goodness parameter

I have to copy the CASA script to produce the images

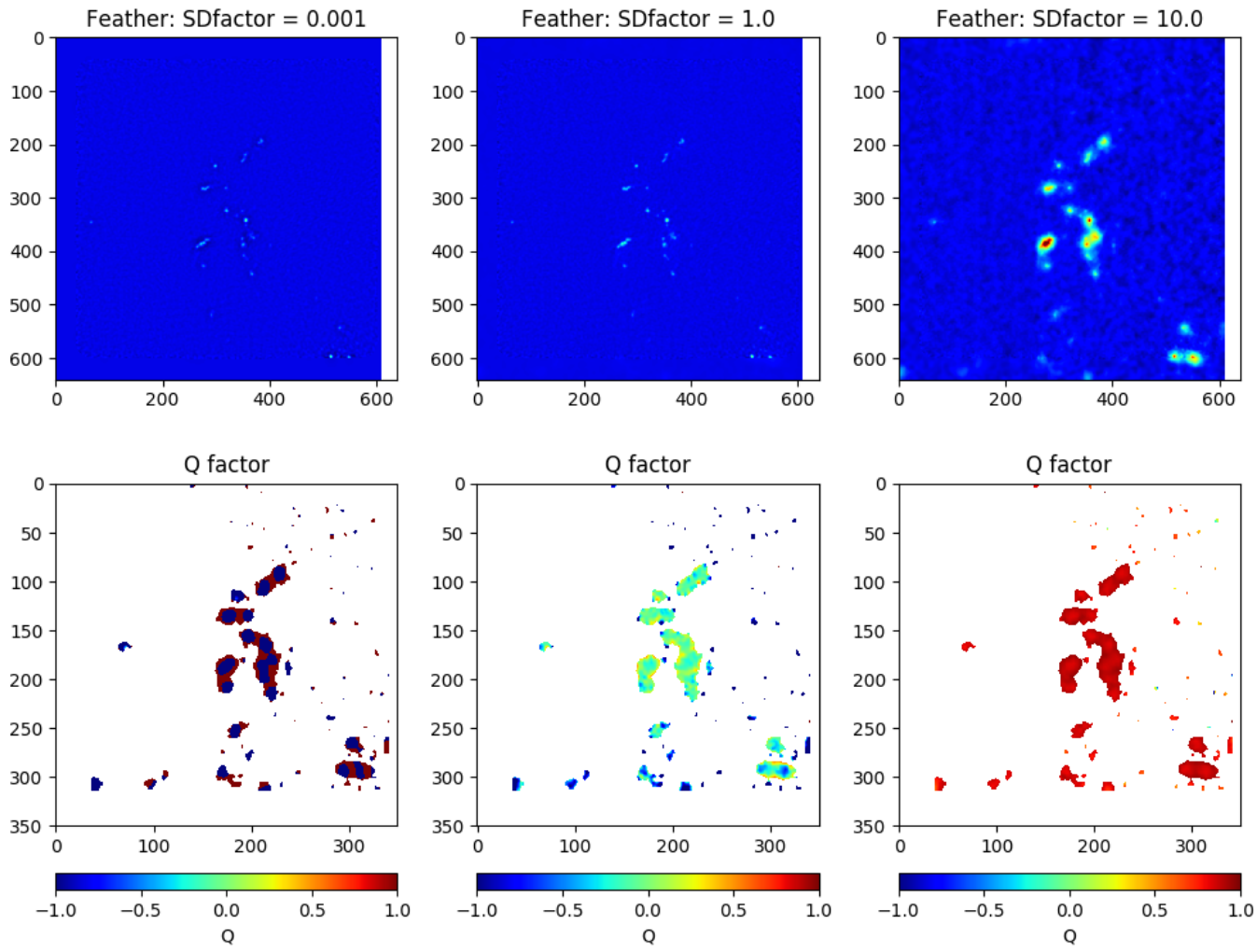


Figure 13: Quality parameter (Q) for different deconvolution + feathering combinations: Feathering with *SDfactor* (1) 0.001 (almost interferometric only data), (2) 1.0, and (3) 10.0.

3.4 Power spectrum comparison

In Progress...

I have to copy the CASA script to produce the images

Need to implement new scrips from Adam.

3.5 Further tests

4 Additional notes, comments, challenges encountered

Error1:

Organize as you wish. Please, if you discovered an issue during this workshop, document it as an issue in the Github.

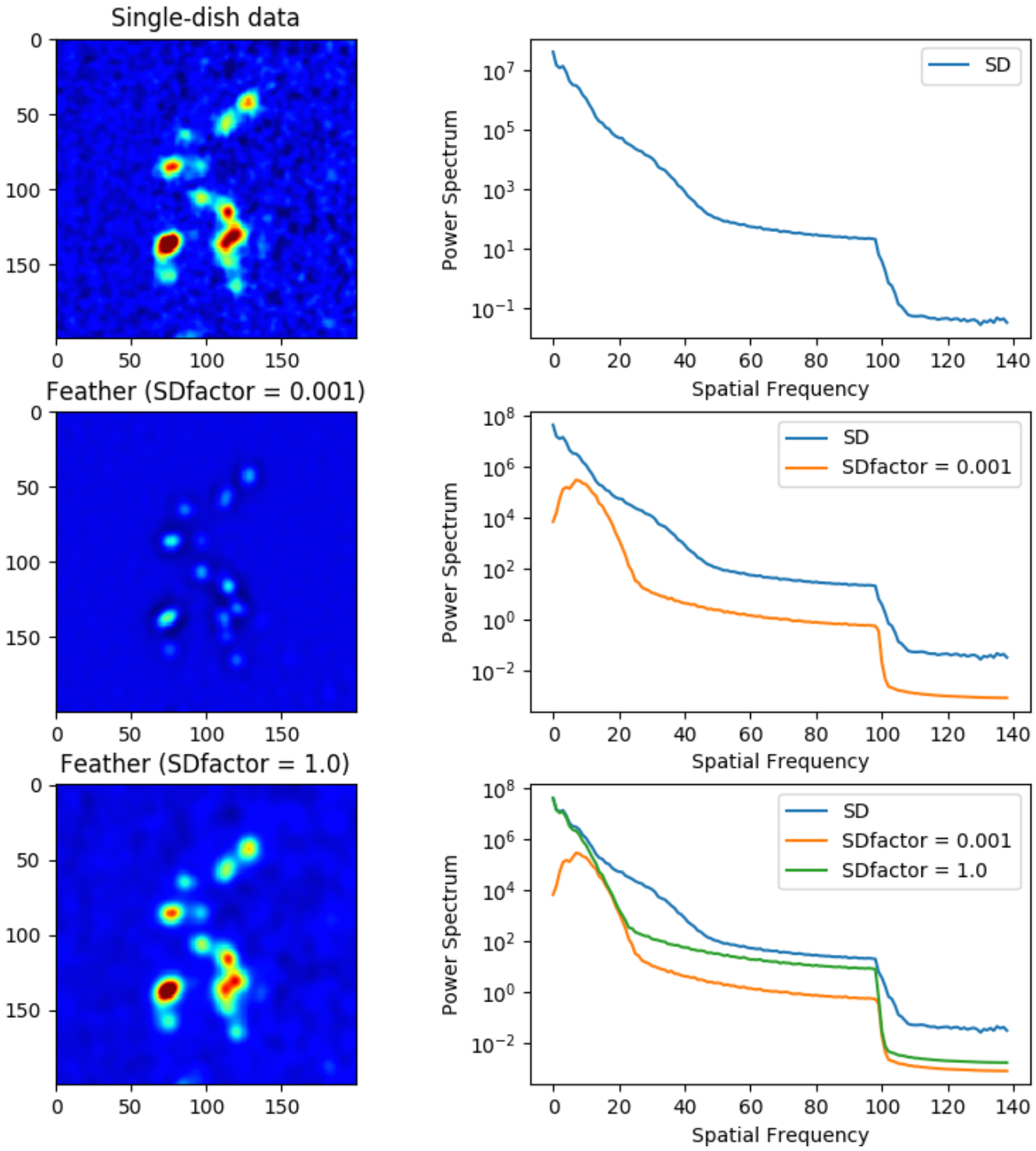


Figure 14: Power-Spectrum for the (Top row) single-dish image in comparison to different deconvolution + feathering combinations with $SDfactor$ (Middle row) 0.001 (almost interferometric only data) and (Lower row) 1.0 SD-factors

5 Appendix: Code

```
def gauss_beam_abg(amp,parms_abg,x,y):
    a = parms_abg[0]
    b = parms_abg[1]
    g = parms_abg[2]
    r = a*(x**2) + b*x*y + g*y**2
    B = amp*np.exp(-r)
    return B
```