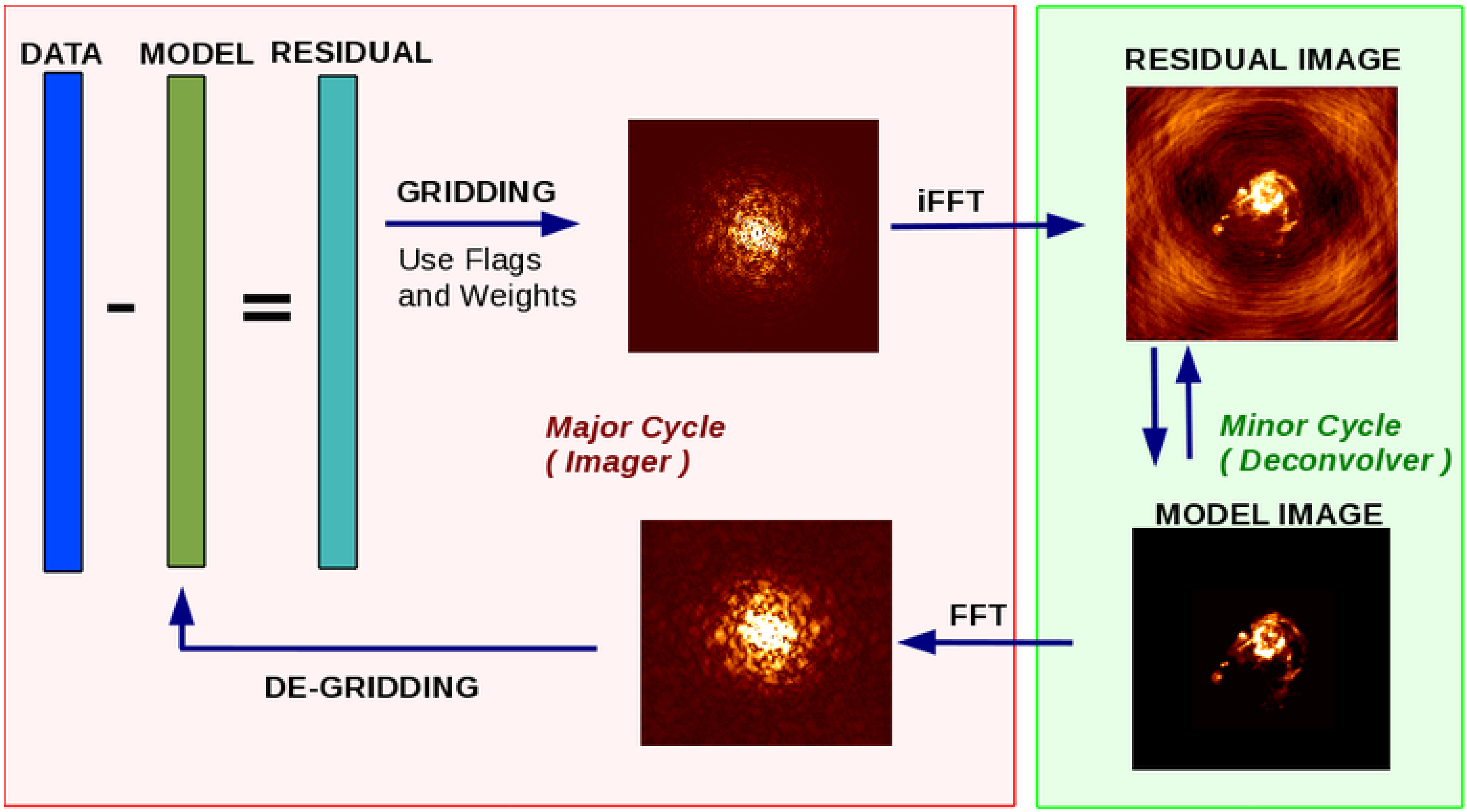
The sdintimaging task (D.Petry, Oct 2024)

sdintimaging(vis, usedata='sdint', sdimage='', sdpsf='', sdgain=1.0, dishdia='',
selectdata=True, field='', spw='', timerange='', uvrange='', antenna='', scan='', observation='',
intent='', datacolumn='corrected', imagename='', imsize=100, cell='"1arcsec"', phasecenter='',
stokes='I', projection='SIN', startmodel='', specmode='mfs', reffreq='', nchan=-1, start='', width='',
outframe='LSRK', veltype='radio', restfreq='', interpolation='linear', perchanweightdensity=True,
gridder='standard', facets=1, psfphasecenter='', wprojplanes=1, vptable='', mosweight=True,
aterm=True, psterm=False, wbawp=True, cfcache='', usepointing=False, computepastep=360.0,
rotatepastep=360.0, pointingoffsetsigdev='', pblimit=0.2, deconvolver='hogbom', scales='',
nterms=2, smallscalebias=0.0, restoration=True, restoringbeam='', pbcor=False, weighting='natural',
robust=0.5, noise='1.0Jy', npixels=0, uvtaper=[''], niter=0, gain=0.1, threshold=0.0, nsigma=0.0,
cycleniter=-1, cyclefactor=1.0, minpsffraction=0.05, maxpsffraction=0.8, interactive=False,
fullsummary=False, nmajor=-1, usemask='user', mask='', pbmask=0.0, sidelobethreshold=3.0,
noisethreshold=5.0, lownoisethreshold=1.5, negativethreshold=0.0, smoothfactor=1.0,
minbeamfrac=0.3, cutthreshold=0.01, growiterations=75, dogrowprune=True, minpercentchange=-1.0,
verbose=False, fastnoise=True, restart=True, calcres=True, calcpsf=True)

- available since CASA 6.1
- CASA 6.6.0: latest round of improvements
- based on (t)clean and feather
- main purpose:
  combination of interferometric and single dish data

sdintimaging (D.Petry, Oct 2024)

(t)clean (operates on INT data (visibilities))



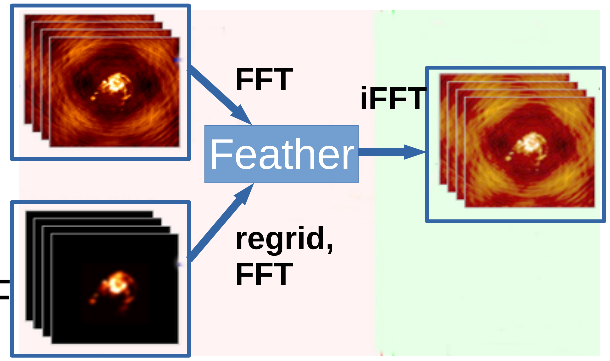
iterative deconvolution building a model and using known PSF

sdintimaging (D.Petry, Oct 2024)

feather (operates on SD and INT image)

INT image/cube

SD image/cube, SD beam/PSF



Feathered image/cube

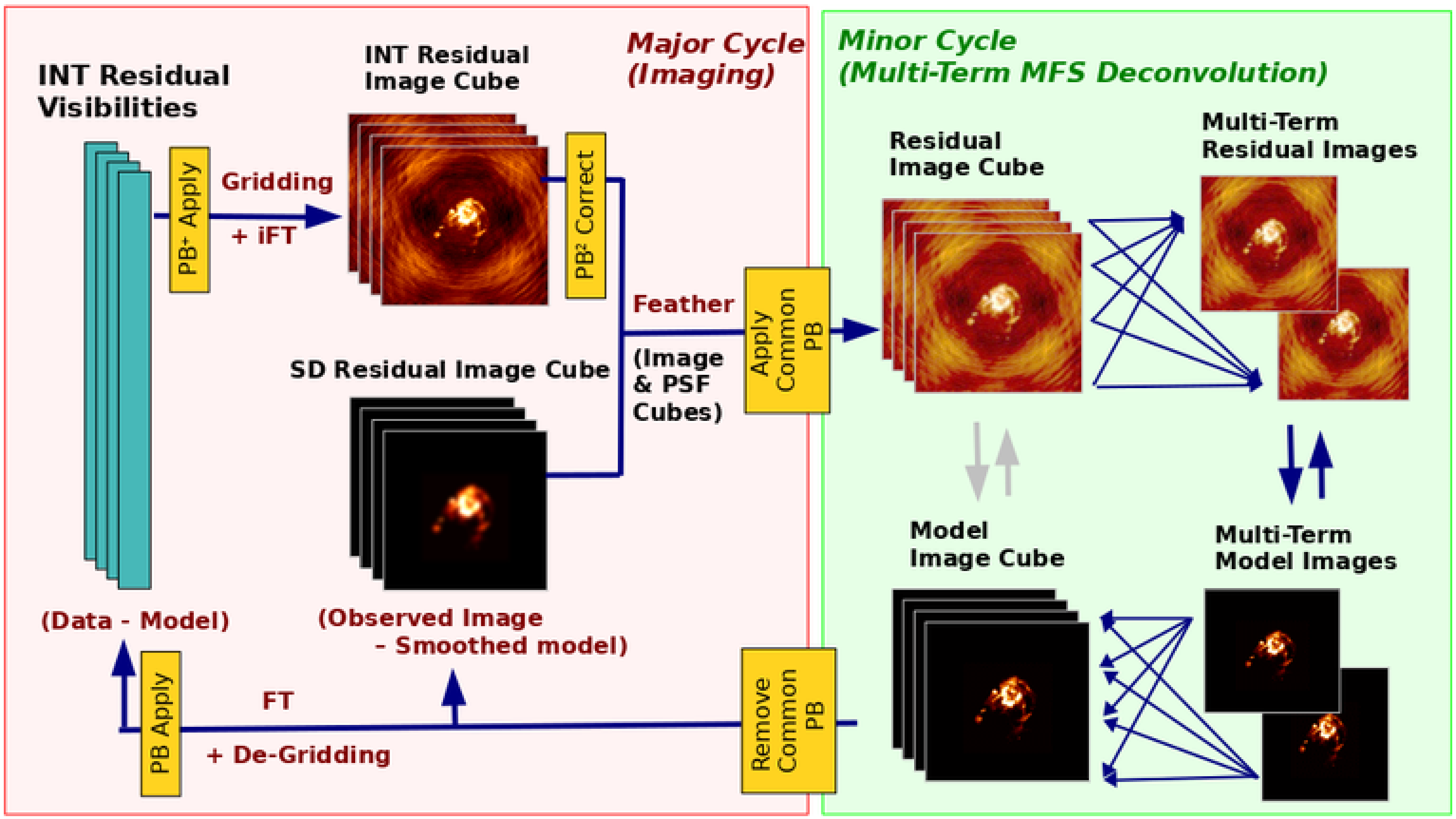
1) Regrid SD image to temporary copy matching resolution of INT image
2) Fourier transform each image to uv plane to obtain *FT(SDimage), FT(INTimage)*
3) Fourier transform the SD beam to obtain *FT(SDbeam)*
4) multiply FT(INTimage) by (1 – FT(SDbeam)) to downweight INT shorter spacings
5) Scale FT(SDimage) by the volume ratio of INT restoring beam and SD beam
6) Add results of (4) and (5) and transform back to the image plane.

No iteration. No deconvolution. INT data already deconvolved.
(In CASA implementation, there is a parameter “***sdfactor***”
which scales the SD image in order to adjust the flux scale
if necessary. **This doesn’t exist in sdintimaging.**)

See nice paper by Bill Cotton https://ui.adsabs.harvard.edu/abs/2017PASP..129i4501C/abstract

sdintimaging (D.Petry, Oct 2024)

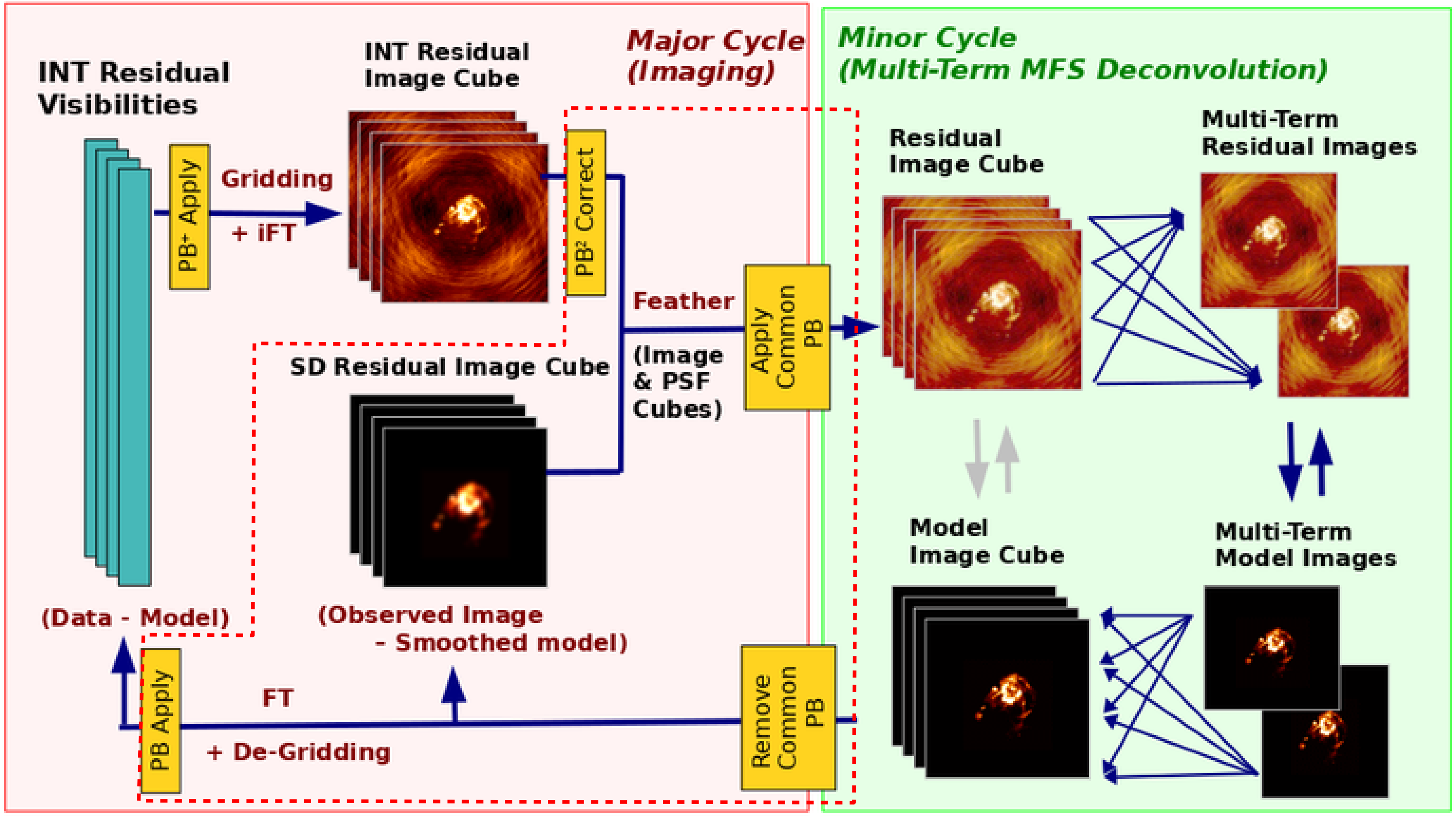
sdintimaging – marry tclean with feather



Description of the SDINT algorithm published in **Rau, Naik & Braun, 2019, AJ, 158, 3**
https://iopscience.iop.org/article/10.3847/1538-3881/ab1aa7

sdintimaging (D.Petry, Oct 2024)

sdintimaging – marry tclean with feather



Description of the SDINT algorithm published in Rau, Naik & Braun, 2019, AJ, 158, 3
https://iopscience.iop.org/article/10.3847/1538-3881/ab1aa7

sdintimaging (D.Petry, Oct 2024)

SDINT Algorithm for spectral cube imaging (from Rau et al. 2019)

```
INITIALIZE IMAGERS AND DECONVOLVER:   IntCube = Imager(gridder = "interferometer")
                                      SDCube = Imager(gridder = "singledish")
                                      JointCube = Imager(deconvolver = "multiscale")
MAKE PSFS:                            IntCube.MakePSF()
                                      SDCube.MakePSF()
                                      JointCube.psf = Feather(IntCube.psf, SDCube.psf)
MAKE INITIAL IMAGES:                  IntCube.MakeRES()
                                      SDCube.MakeRES()
                                      IntCube.res = IntCube.res ÷ IntCube.PB
                                      JointCube.res = Feather(IntCube.res, SDCube.res)
                                      JointCube.res = JointCube.res × IntCube.PB
  repeat
    DECONVOLVE PER CHANNEL PSF:          JointCube.deconvolve()
    UPDATE RESIDUAL IMAGES:              IntCube.MakeRES(JointCube.mod)
                                         SDCube.MakeRES(JointCube.mod/IntCube.PB)
                                         IntCube.res = IntCube.res ÷ IntCube.PB
                                         JointCube.res = Feather(IntCube.res, SDCube.res)
                                         JointCube.res = JointCube.res × IntCube.PB
  until Convergence criteria are satisfied
RESTORE MODEL AND PB-CORRECT:         JointCube.restore()
                                      JointCube.image = JointCube.image ÷ IntCube.PB
```

sdintimaging (D.Petry, Oct 2024)

sdintimaging

- can be used for
  a) joint reconstruction of interferometer (INT) and single dish (SD) data – **usedata = ‘sdint’**
  b) cube and wideband multi-term deconvolution of SD data (experimental, needs INT data for templates) – **usedata = ‘sd’**
  c) for gridder=’mosaic’ or ‘awproject’: perform wideband mosaic on INT data similar to tclean, but with conjugate-pb correction in the image domain (experimental) – **usedata = ‘int’**

- shares most parameters with tclean

- additional parameters mostly concern handling of SD image and PSF

sdintimaging (D.Petry, Oct 2024)

sdintimaging – *advantages over other combination methods*

- **joint deconvolution of INT and SD data:** easier to construct a model which takes into account both SD and INT information

- for comparable INT and SD noise levels, result is similar to generating visibilities from SD data and joint tclean with INT data (TP2Vis)

- in case of significantly different INT and SD noise levels, **matching noise levels is simpler in sdintimaging** than in TP2Vis (use parameter ***sdgain***)

sdintimaging (D.Petry, Oct 2024)

Understanding the parameter “sdgain”

- sdgain is essentially the **relative weight of the SD data w.r.t. the INT data**
- can be derived from the RMS of the individual SD and INT images
- If we measure the sensitivity of the SD and INT images separately (but based on the same spectral range), then we get two values
  RMS_sd and RMS_int (e.g. in units of Jy/sr)
  The canonical combination of, e.g., the two flux measurements from the two datasets (F_sd and F_int) is then the weighted mean

  (1) $F = 1/W \cdot (w_{sd} \cdot F_{sd} + w_{int} \cdot F_{int})$

  where $W = w_{sd} + w_{int}$
  $w_{tp} = (1/RMS_{sd})^2$
  $w_{int} = (1/RMS_{int})^2$

sdintimaging (D.Petry, Oct 2024)

Understanding the parameter "sdgain" (cntd.)

(from prev. slide) $F = 1/W \cdot (w\_sd \cdot F\_sd + w\_int \cdot F\_int)$

where $W = w\_sd + w\_int$

$w\_tp = (1/RMS\_sd)^2$

$w\_int = (1/RMS\_int)^2$

- sdgain is then defined via

(2) $F = 1/(sdgain+1) \cdot (sdgain \cdot F\_sd + F\_int)$

- setting (1)=(2) and solving for sdgain gives

(3) **sdgain =** $w\_sd/w\_int =$ $\mathbf{(RMS\_int/RMS\_sd)^2}$

i.e. the ratio of the INT and SD sensitivities squared!

The better the SD sensitivity (relatively), the larger sdgain.

sdintimaging (D.Petry, Oct 2024)

sdgain calculation example:

The ALMA Northern SMC CO survey (2017.A.00054.S), Band 6

RMS_int = 0.1 Jy/beam in 2.4 MHz channel for beam area 26.4 $asec^2$

RMS_sd = 0.64 Jy/beam in 2.4 MHz channel for beam area 452 $asec^2$

$sdgain = (RMS\_int[Jy/sr]/RMS\_sd[Jy/sr])^2 = (0.1/0.64 \cdot 452/26.4)^2 = 7.2$

For this test dataset, values of sdgain around 5.0 have given very reasonably looking combined images.

NOTE: the correct sdgain can be >> 1.0

sdintimaging (D.Petry, Oct 2024)

CASA
Common Astronomy Software Applications

A look at the documentation:
https://casadocs.readthedocs.io/en/stable/api/tt/casatasks.imaging.sdintimaging.html

- Data selection applies to the INT data and is the same as in tclean.

- **sdimage**: the input SD image (can be a cube)

- **sdpsf**: (optional) PSF images for each channel of the sdimage cube.
  If omitted: PSF will be derived from beam stored with sdimage
  (assuming Gaussian shape).

- **imsize, cell, phasecenter**: as in tclean
  (the parameters of the sdimage will be adjusted if they don’t match
  by regridding)

- **start, width, nchan**: sdimage must have same spectral grid!

sdintimaging (D.Petry, Oct 2024)

A look at the documentation (cntd.):

- **sdimage** must have axes RA, DEC, Stokes, Freq

- If INT or SD data is flagged in a channel, this channel is flagged in the output cube

- for **specmode** ‘cube’, use of **deconvolver**=’multiscale’ is recommended

- **specmode** ‘mfs’ only works with **nterms**=1 and **deconvolver**=’mtmfs’, **reffreq** must be set to a value within the spectral range!

- **gridder** settings ‘mosaic’ and ‘awproject’ are recommended since they do a full pb correction

- Parameter **interactive** works as for tclean

sdintimaging (D.Petry, Oct 2024)

Output Images

<imagename>.sd.cube.{image,psf}
Image cubes onto which the input Single Dish image and psf cubes are regridded.

<imagename>.sd.cube.{model,residual}
The model image cube that is subtracted from the SD image to make the SD residual

<imagename>.int.cube.{residual, psf, sumwt,weight,pb)
Image cubes made from only the interferometer data

<imagename>.int.cube.{model}
Cube model image used for model prediction and residual calculation.

<imagename>.joint.cube.{residual, psf}
Feathered cubes for the residual and psf.

<imagename>.joint.multiterm.{residual,psf}.{tt0,tt1[,tt2]}
Multi-term residual images and spectral PSFs constructed from feathered cubes.

Output Images (cntd.)

<imagename>.joint.cube.{image, sumwt, weight, pb, model, mask, image.pbcor}

The standard output images as for tclean in the case of a single-term specmode.

<imagename>.joint.multiterm.{image, sumwt, weight, pb, model, mask, alpha, pbcor}
with {.tt0, .tt1, .tt2 } extensions as appropriate.

The standard output images as for tclean in case of a multi-term specmode.

sdintimaging (D.Petry, Oct 2024)

Example: ALMA NGC4945 Spectral Cube Imaging : 7M + SD

from project 2021.1.00783.S

7M data: 3 EBs, mosaic with 12 pointings, recommended cell size 1.2 arcsec, imsize 144 x 144 pixels

Our example line is contained in channels 500~900 in SPW 20.

SD data: *corresponding SPW is 21* (SD and INT don't use same SPW numbering!), we produce the matching image using sdintimaging

NOTE: before CASA 6.6.0, SD image needed to have per-plane restoring beams! Per-plane restoring beams may be added to an existing image cube using ia.open(), a loop over channels with ia.setrestoringbeam(..), and ia.close(). Since CASA 6.6.0, this is done automatically.

sdintimaging (D.Petry, Oct 2024)

Example: NGC4945 Spectral Cube Imaging : 7M + SD *(cntd.)*

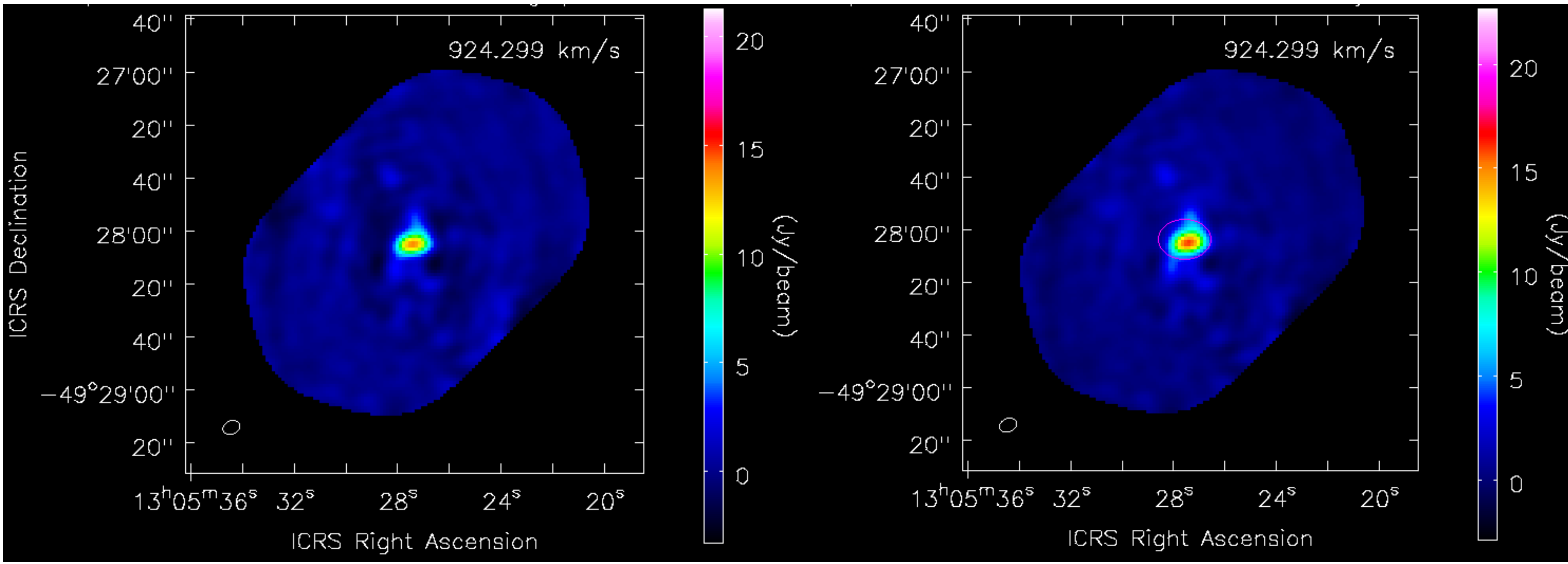
Run sdintimaging (under CASA 6.6.1)

```
sdintimaging(usedata='sdint',
        sdimage='NGC4945_sci.spw21.cube.I.manualsd500',
        sdpsf='', # we let sdintimaging generate that
        dishdia=12., # SD dish diameter in meters
        sdgain=1.,
        vis = thevis,
        imagename = 'NGC4945_sci.spw20.mos.7mTP.cube.I.manual-combined4',
        field = 'NGC4945', intent = 'OBSERVE_TARGET#ON_SOURCE',
        phasecenter = 2, stokes = 'I',
        spw = '20', start = 500, nchan = 400, width = 1, outframe = 'LSRK',
        specmode = 'cube',
        #reffreq = '230.08991GHz', # reffreq not needed for cube
        imsize = [144, 144], cell = '1.2arcsec',
        gridder = 'mosaic', deconvolver = 'multiscale', scales=[0, 3, 5],
        niter = 1000, weighting = 'briggs', robust = 0.5, perchanweightdensity=True,
        mask = 'circle[[72pix,72pix],36pix]', # automasking is also possible
        pbcor = True, threshold = '0.1mJy',
        restoringbeam = 'common', interactive = False )
```

sdintimaging **(D.Petry, Oct 2024)**

Example: NGC4945 Spectral Cube Imaging : 7M + SD *(cntd.)*

Resulting image (peak channel)



7M INT data only (tclean)

7M INT + SD data (sdintimaging)

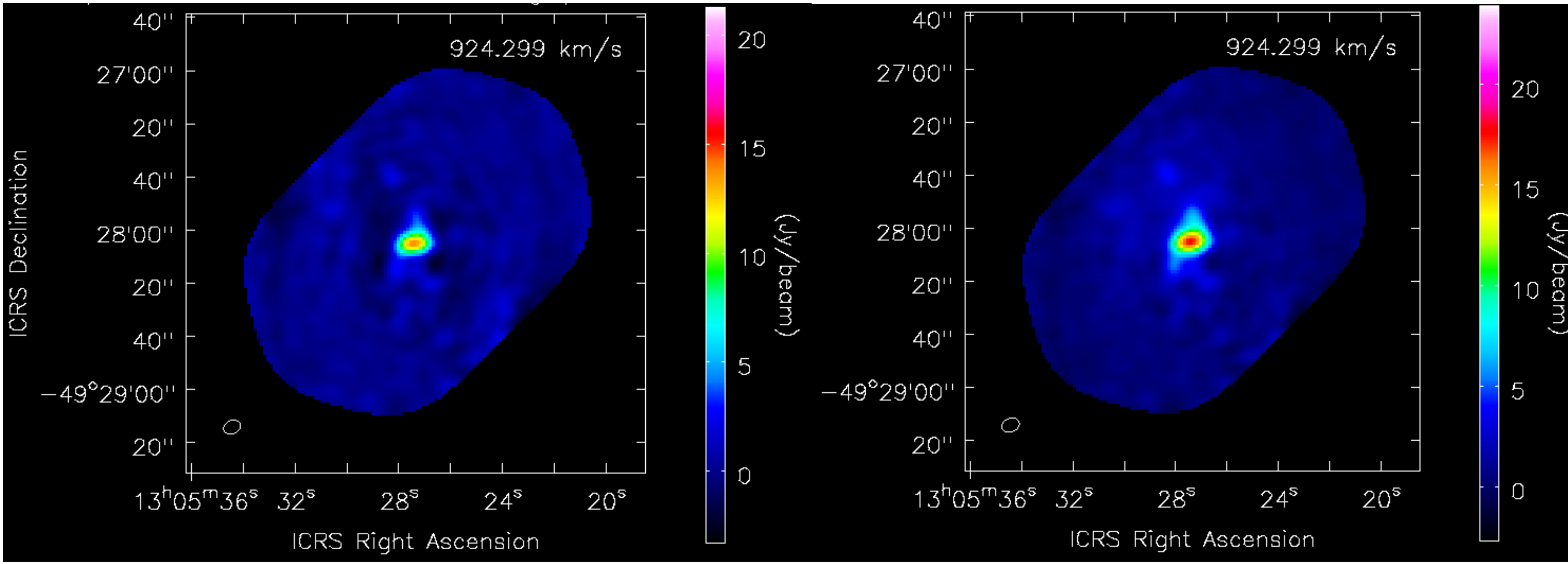
Negative troughs *partially* eliminated, higher peak flux.
Can still try to determine **sdgain** more accurately
(so far we chose the default sdgain=1.0)

sdintimaging (D.Petry, Oct 2024)

Example: NGC4945 Spectral Cube Imaging : 7M + SD *(cntd.)*

Resulting image (peak channel)

Now using sdgain=2.0



7M INT data only (tclean)

7M INT + SD data (sdintimaging)

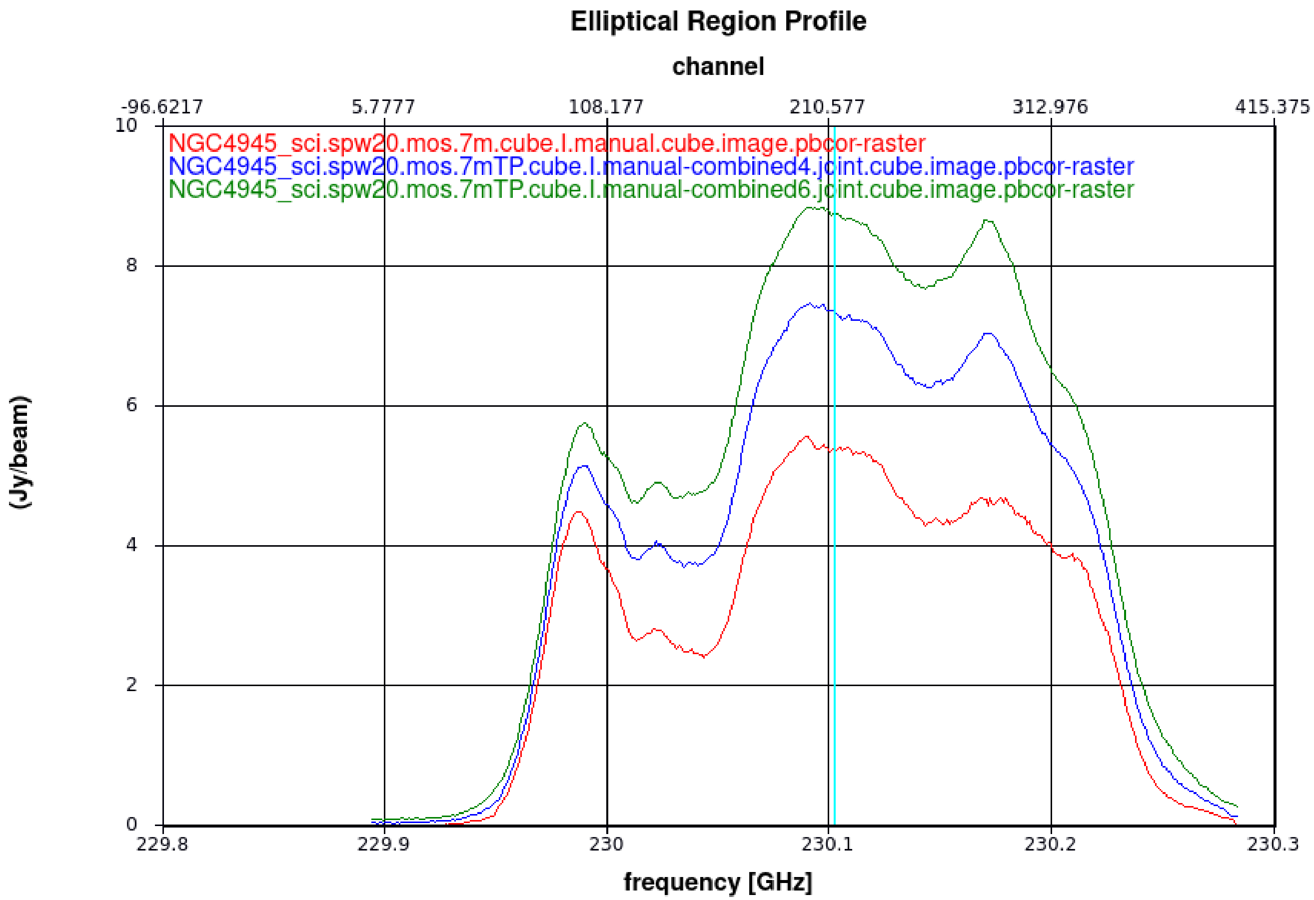
Negative troughs mostly eliminated,
further increased peak flux.

sdintimaging (D.Petry, Oct 2024)

CASA
Common Astronomy Software Applications

Example: NGC4945 Spectral Cube Imaging : 7M + SD *(cntd.)*

Spectral profile of the central region: addition of SD makes major difference!
But also correct choice of sdgain is critical!



7M INT+SD data (sdgain=2)

7M INT+SD data (sdgain=1)

7M INT data only (tclean)

sdintimaging (D.Petry, Oct 2024)

Summary

- sdintimaging is a major contender for ALMA standard data combination

- CASA 6.2 contained the first reasonably debugged version of sdintimaging

- CASA 6.6.0 had further improvements, now also single-channel input works well

- The optimal *sdgain calculation* is still being investigated
  and might be automated in a future version.

- In CASA 6.6.1, sdintimaging might still produce many warning messages like
  `“You are regridding an image whose beam is not well sampled ...”`
  These can be ignored.

- **Feedback and example cases from general users are welcome!**
  Send ALMA data reduction helpdesk ticket, attention of Dirk Petry.