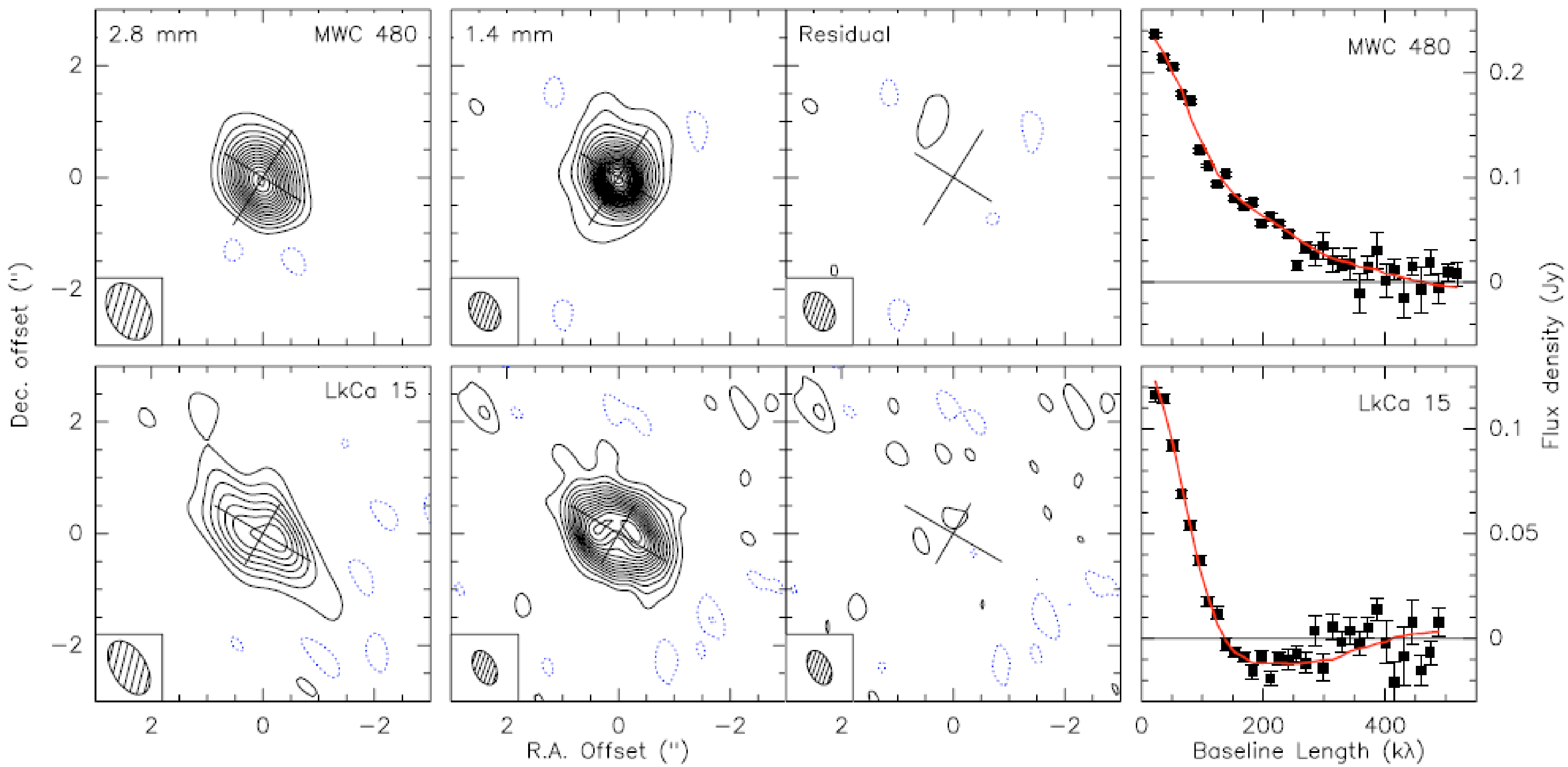
Self-calibration: about the implementation in GILDAS

Vincent Piétu
IRAM

About an interferometer sensitivity

- One usually considers only the noise equation to assess the feasibility of an observation.
- However there are some cases where the noise in the image is somewhat higher than the thermal noise would predict.
- In practice, the dynamic range (ratio of the brightness peak over the noise) was usually limited to 30-80 for PdBI.

Example



the resolution is a factor lower. Based on the integration time, system noise, and measured efficiencies, the expected (thermal) noise level was 0.7 mJy/beam at 220 GHz. However, the dynamic range is limited by phase noise. This results in an effective noise of 0.9 mJy/beam for LkCa 15 and 2.0 mJy/beam for MWC 480. At 110 GHz, the noise is 0.3 mJy/beam, so essentially thermal.

Calibrations

- When we calibrate data, we use the calibrator data, observed every t hour, so we account for effect with T > 2*t (Nyquist sampling). Attempts to fit faster components will result in aliasing, i.e. an increased, calibration-induced, noise.
- This is why one should not use undersampled calibration curve.
- Effects happening on shorter time scale will appear as off the calibration curve, and we can only quantify their magnitude by computing the rms w.r.t. the calibration curve.
- Amplitude: usually not much residuals.
- Phase: atmospheric phase is barely calibrated out (can be increased using fast switching, used at ALMA). Observatory try to observe in reasonnable conditions, but having 30 degrees rms at the highest frequency and larger configurations is challenging.
- One possible solution is to calibrate the instrument on the source itself if it bright enough.

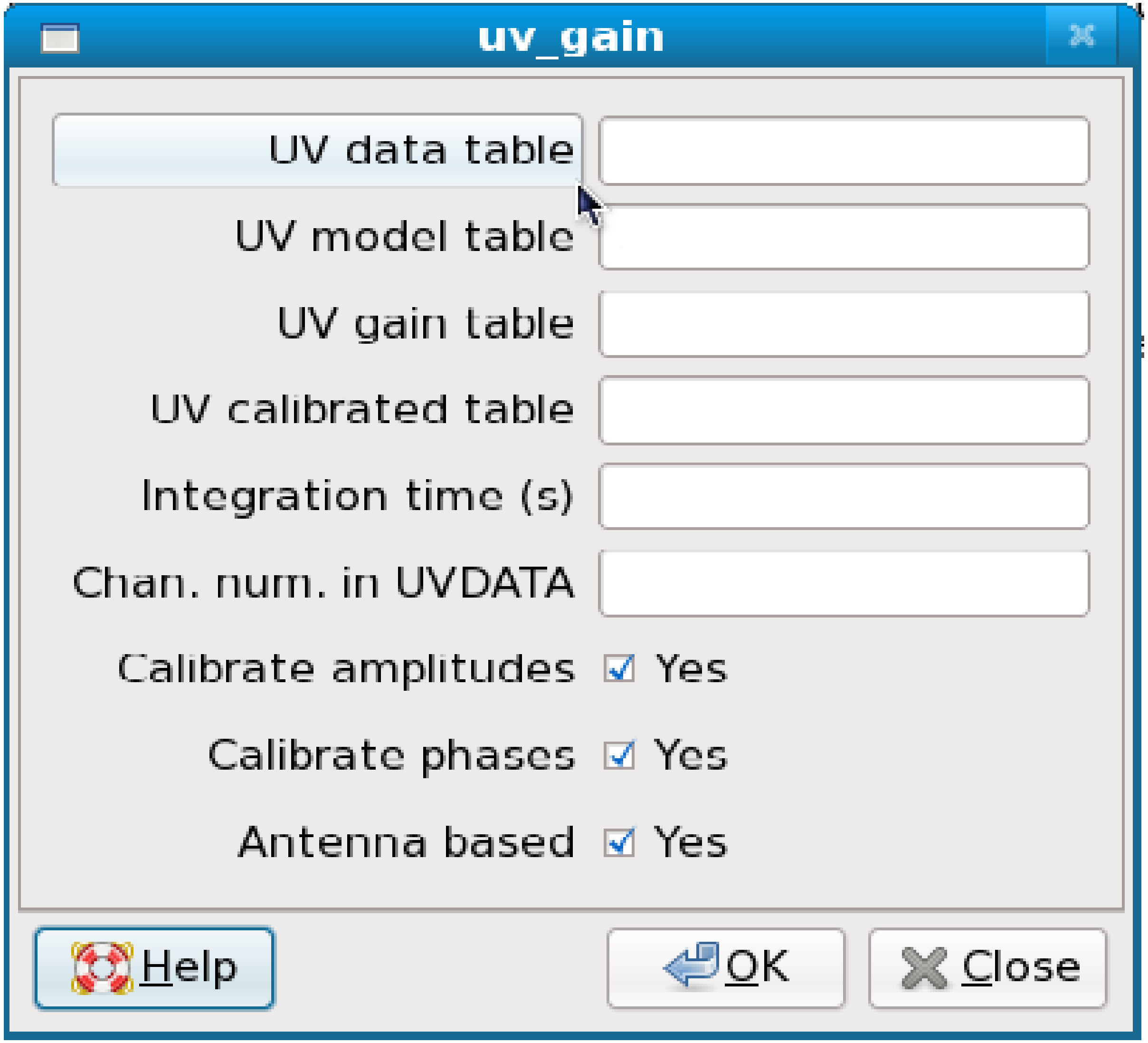
Selfcal in GILDAS: algorithm

- Baseline self-calibration:
  - Provide a model
  - Divide the data visibility by the model visibility -> gain
  - Correct the data by the gain
  - i.e. get the model within the thermal noise
  - Except in those specific cases (e.g. use continuum emission to calibrate lines in planets) not very useful.

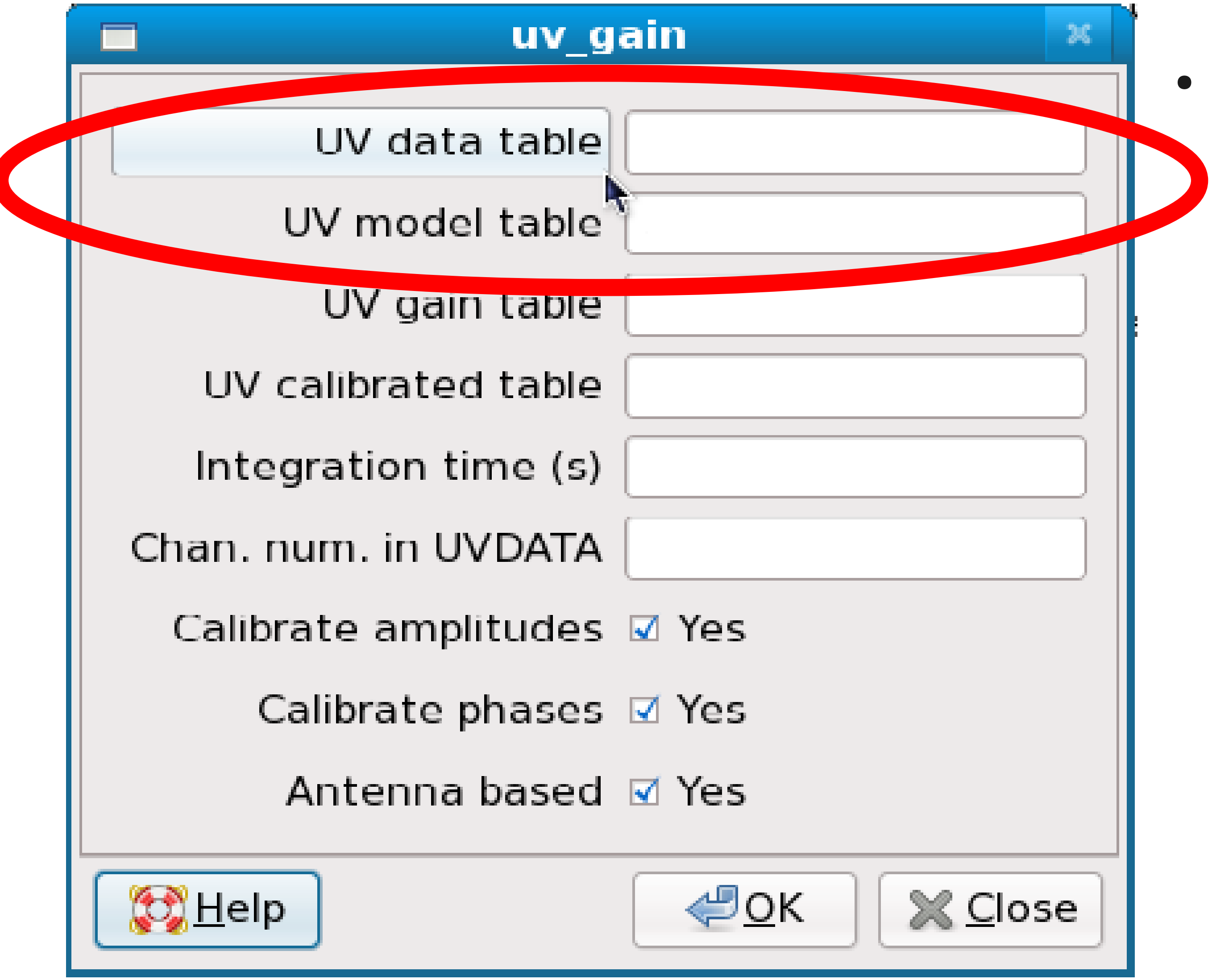
Selfcal in GILDAS: algorithm

- Antenna self-calibration:
  - Start as for baseline selfcalibration (divide visibility by model) -> baseline gain
  - Average the baseline gain in time (running average)
  - Find a good reference antenna (that minimize uv)
  - Factorize per antenna to get antenna gain
  - Apply gain to get corrected data.

uv_gain task in GILDAS

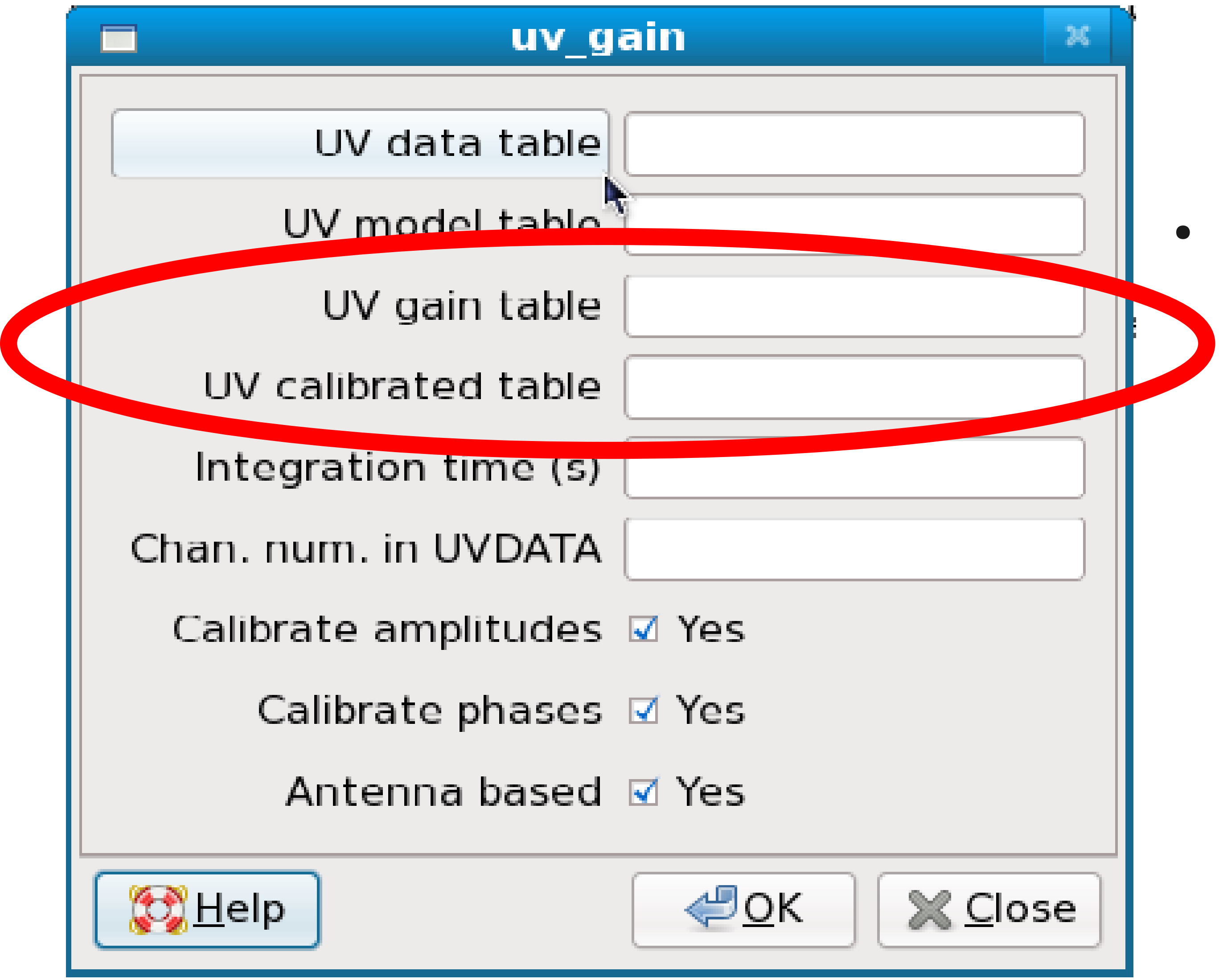


uv_gain task in GILDAS



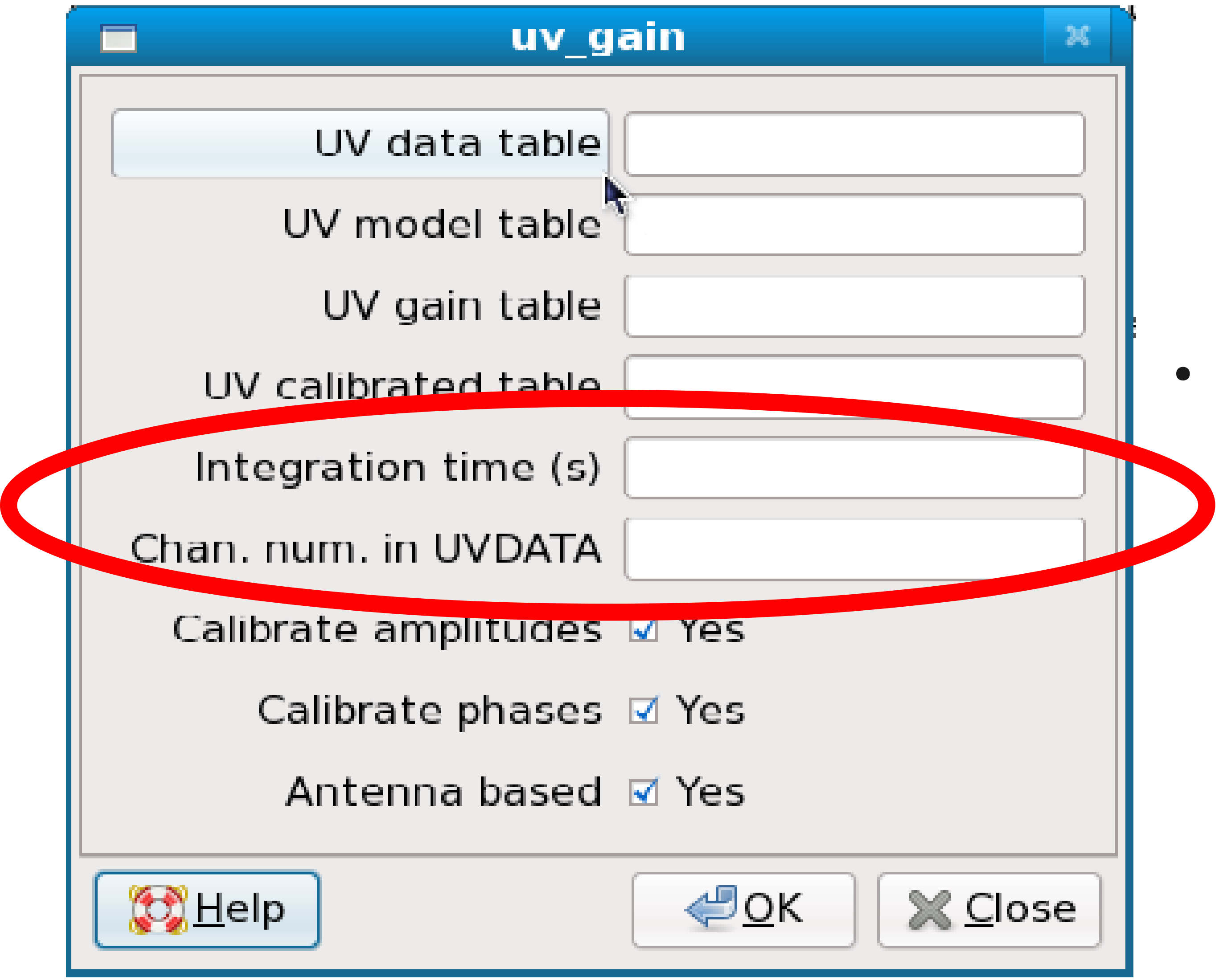
- Inputs:
  - uncorrected data and
  - model (with same uv coverage than data but a single channel)

uv_gain task in GILDAS



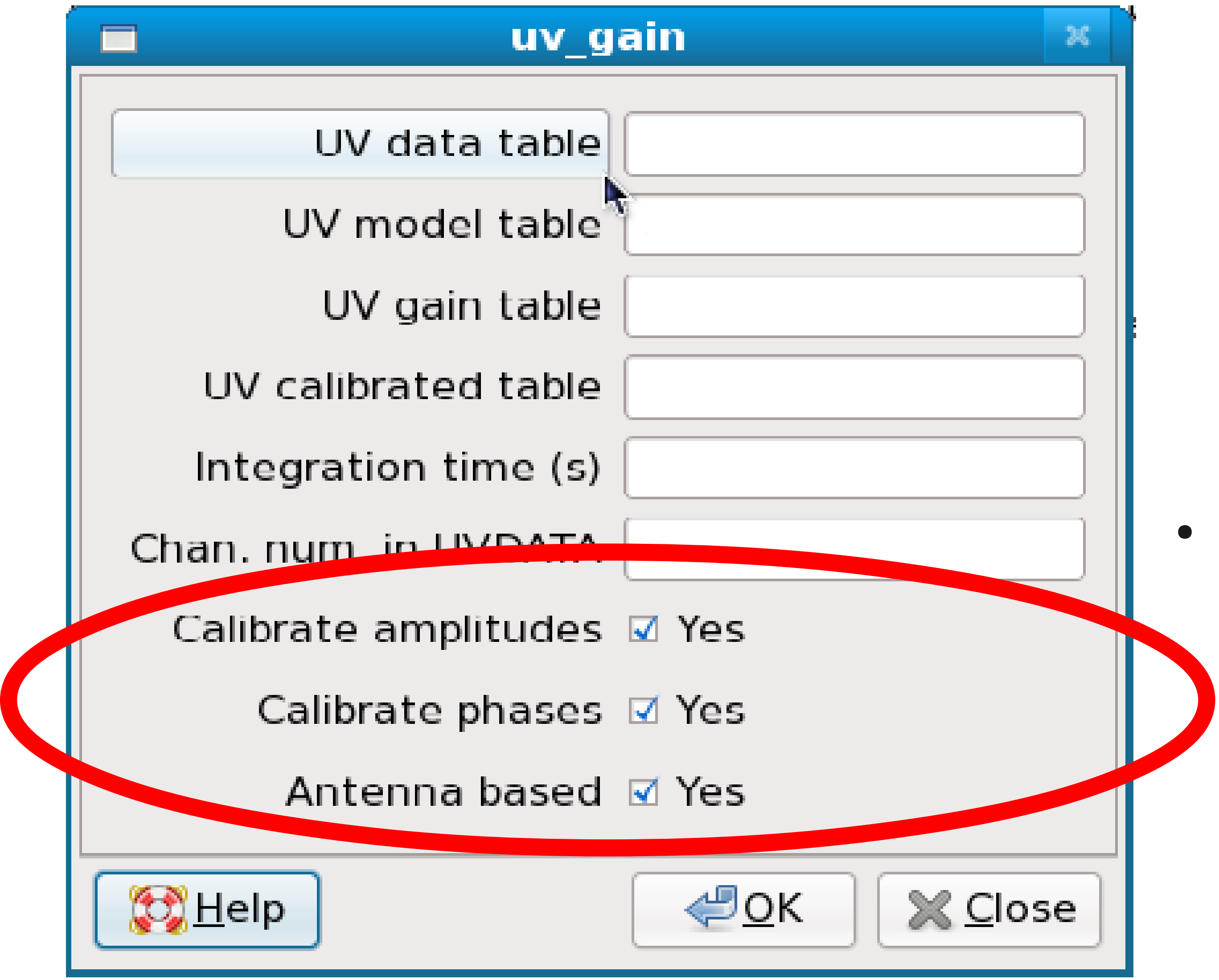
- Outputs:
  - Gain table and
  - corrected data

uv_gain task in GILDAS



- Parameters:
  - Time averaging value
  - Data channel to which to compare the model

uv_gain task in GILDAS



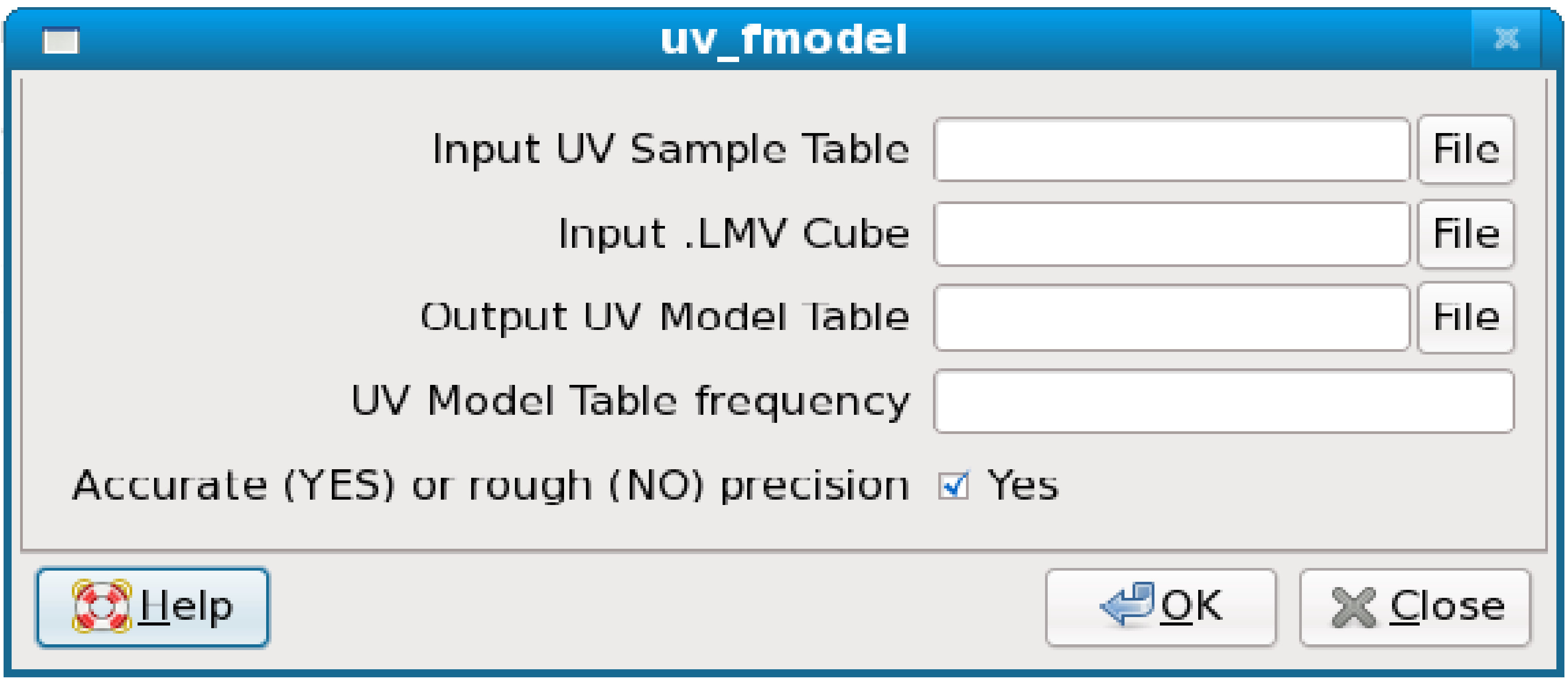
- Control parameters
  - Correct for amplitude
  - Correct for phase
  - Use antenna/baseline algorithm

How to get a model

- You have one
  - Image: run uv_fmodel

How to get a model

- You have one
  - Image: run uv_fmodel



How to get a model

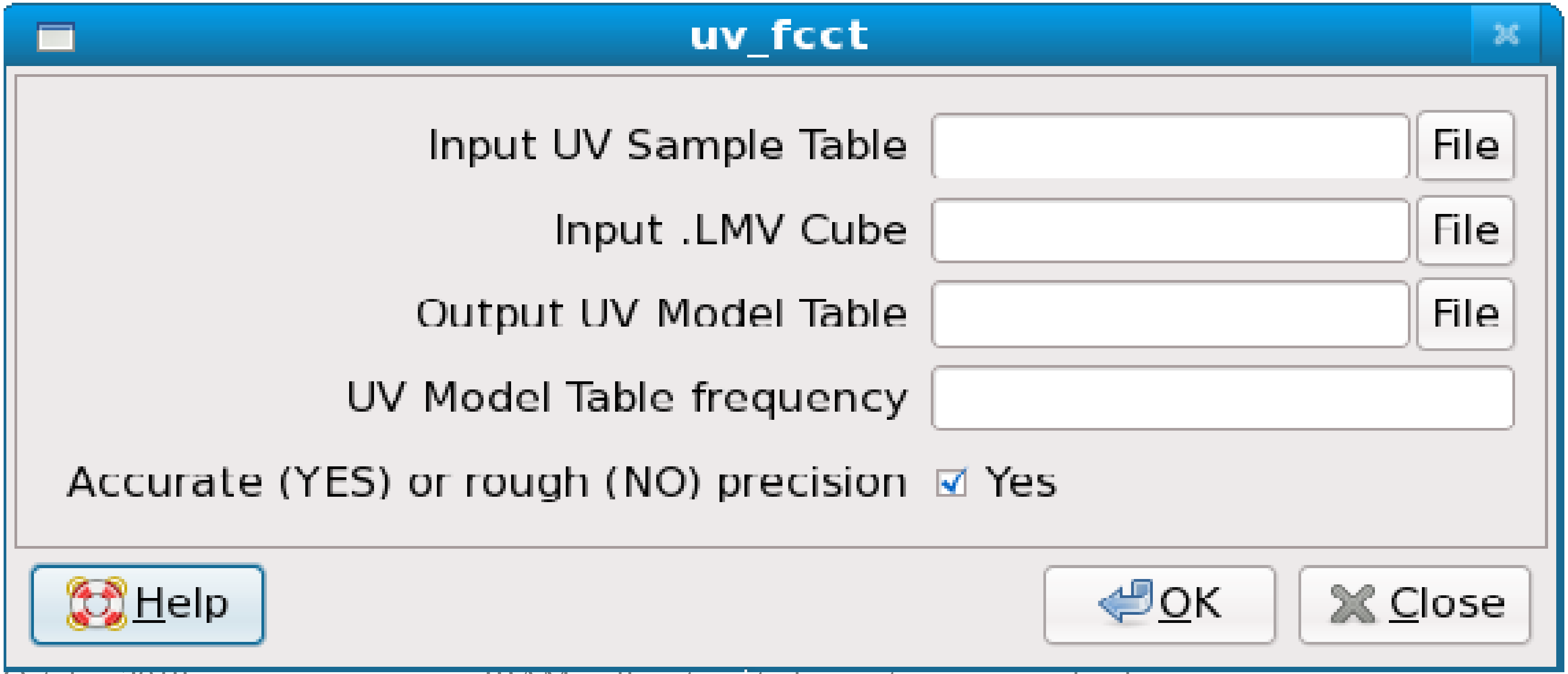
- You have one
  - Image: run uv_fmodel
  - uv model: use it
    - e.g. using uv_fit to get a model

How to get a model

- You have one
  - Image: run uv_fmodel
  - uv model: use it
    - e.g. using uv_fit to get a model
- You do not have one
  - You do still have one
  - You probably cleaned your image
  - So you have a list of clean components
    - Run uv_fcct to generate a model uv table

How to get a model

- You have one
  - Image: run uv_fmodel
  - Uv model: use it
    - e.g. using uv_fit to get a model
- You do not have one



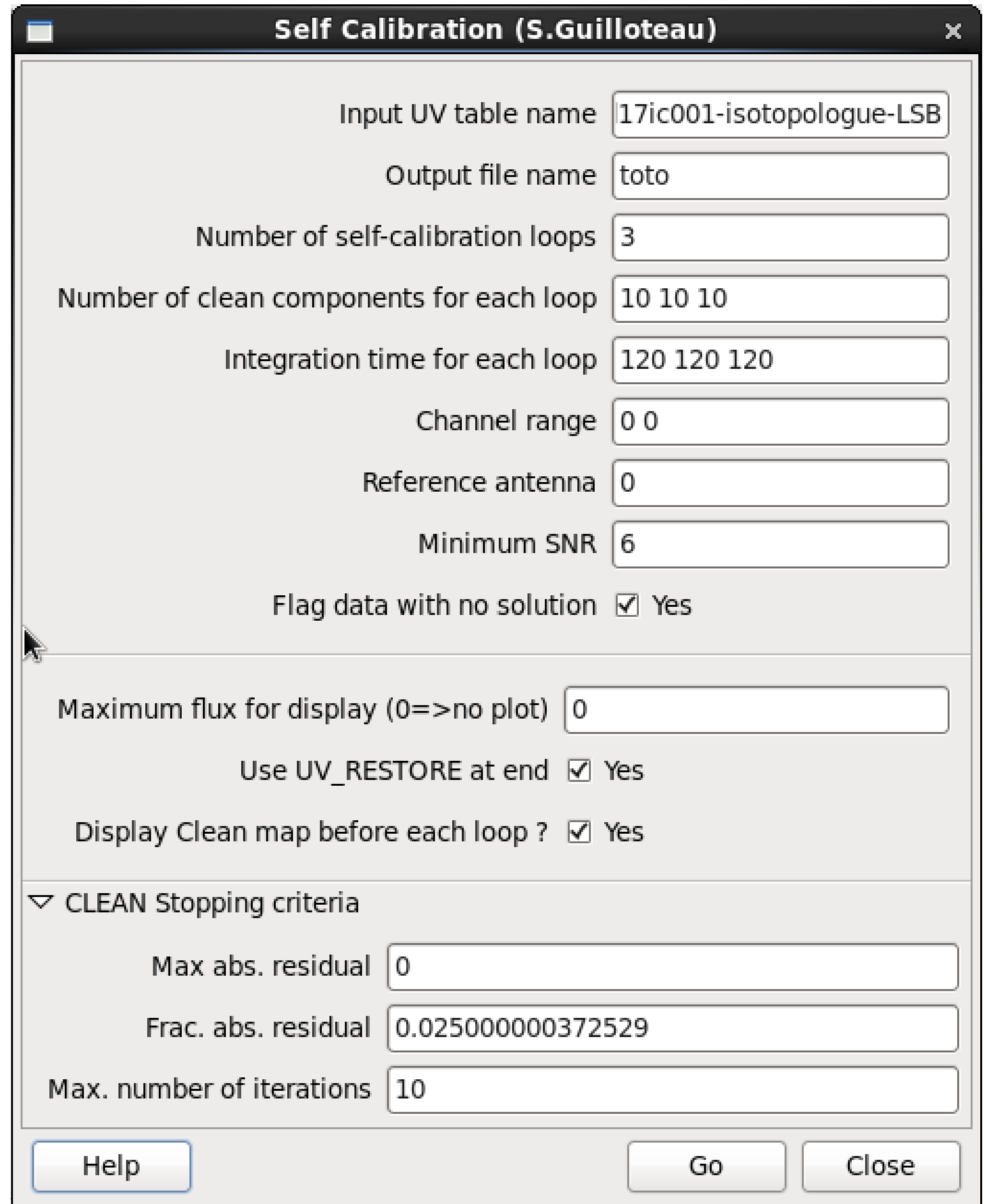
Even simpler: the selfcal procedure

- Built-in GILDAS procedure using uv_fcct and uv_gain tasks
- Called with *go selfcal*
- Can check input parameters with *input selfcal*
- First call creates a structure *SELF%*
  - *self%iname*
  - *self%oname*
  - *self%iter*
  - *self%time ...*
- Does only a phase selfcalibration !

```
MAPPING> go selfcal
 SELFCAL computes and applies self-calibration to a UV Table
   V 2.0: - allows several self-calibration loops in case of complex object
   V 2.1: - allows self-calibrating using a channel range of a line table
   V 2.2: - Control reference antenna Nov-2015

* SELFCAL uses the imaging parameters of UV_MAP
          and the some deconvolution parameters of CLEAN

* Input UV Table is given by SELF%INAME
* Output UV Table and images are specified by SELF%ONAME
     At exit, NAME = SELF%ONAME

   INPUT  UV Table [ .uvt ]
   OUTPUT UV Table [ .uvt ]
   OUTPUT Images [ .lmv-clean , .beam and .lmv ]


   SELF%LOOP  [ 1 ]                 Number of Self Cal loops
   SELF%NITER [  10 ]   Number of selected components
   SELF%TIMES [  120 ]   Integration time for solution
   SELF%CHANNEL  [ 0 0 ]         Channel range
   SELF%REFANT  [ 0 ]           Reference antenna
   SELF%SNAME  [  ]           Solution table

   SELF%FLUX  [ 0 ]                 Maximum flux for display
   SELF%RESTORE [ YES ]            Use UV_RESTORE at end
   SELF%DISPLAY [ YES ]            Display CLEAN image at each loop

Hogbom CLEAN Parameters

   NITER [ 0 ]                         GAIN [ 0.2 ]
   FRES [ 0.025 ]                   ARES [ 0 ]

UV_MAP Parameters
        Map nor shifted neither rotated

        UV_TAPER [ 0 0 0 ]              TAPER_EXPO [ 2 ]
        WEIGHT_MODE [ NATURAL ]         UV_CELL [ 7.5 1 ]
        MAP_SIZE [ 0 0 ]                MAP_FIELD [ 0 0 ]
        MAP_CELL [ 0 0 ]                WCOL [ 0 ]
        MCOL     [ 0 0 ]                CONVOLUTION [ 5 ]
MAPPING>
```

Widgets

- Selfcalibration also accessible through a widget in the menu.



Sensitivity (I)

- Let's consider the classical radiometric formulas:

Baseline sensitivity

$$\sigma_b = \frac{\sqrt{2}kT_{sys}}{\eta_a A \eta_q \sqrt{\Delta\nu\Delta t}}$$

Antenna "efficiency"

$$\mathcal{J} = \frac{2k}{\eta_a A}$$

Baseline sensitivity

$$\sigma_b = \frac{\mathcal{J}}{\eta_c \eta_p} \frac{T_{\mathrm{sys}}}{\sqrt{2\Delta\nu t}}$$

- 2k/A = 15.6 Jy/K (Bure)

Sensitivity (II)

Antenna sensitivity

$$\sigma_g = 2\frac{\sigma_b(\Delta\nu, t)}{S_\nu}\sqrt{\frac{2N-3}{2(N-1)(N-2)}} \simeq 2\frac{\sigma_b}{S_\nu\sqrt{N}}$$

Phase sensitivity

$$\sigma_\phi \ (\text{radians}) \ = \sigma_g \quad \sigma_\phi \ (^\circ) \simeq 57.3\sigma_g$$

- SNR = 1 means ~ 60 deg accuracy on the phase
- SNR = 2 means ~30 deg accuracy on the phase
- SNR = 5 means ~12 deg accuracy on the phase

Typical antenna sensitivity

- Typical values (computed with 6 antennas, 8GHz continuum)

Band	1	2	3	4
Tsys	100	150	250	500
T=45 s	1.2 mJy	2.5 mJy	5 mJy	12.5 mJy
T=120 s	0.7 mJy	1.5 mJy	3 mJy	7.7 mJy

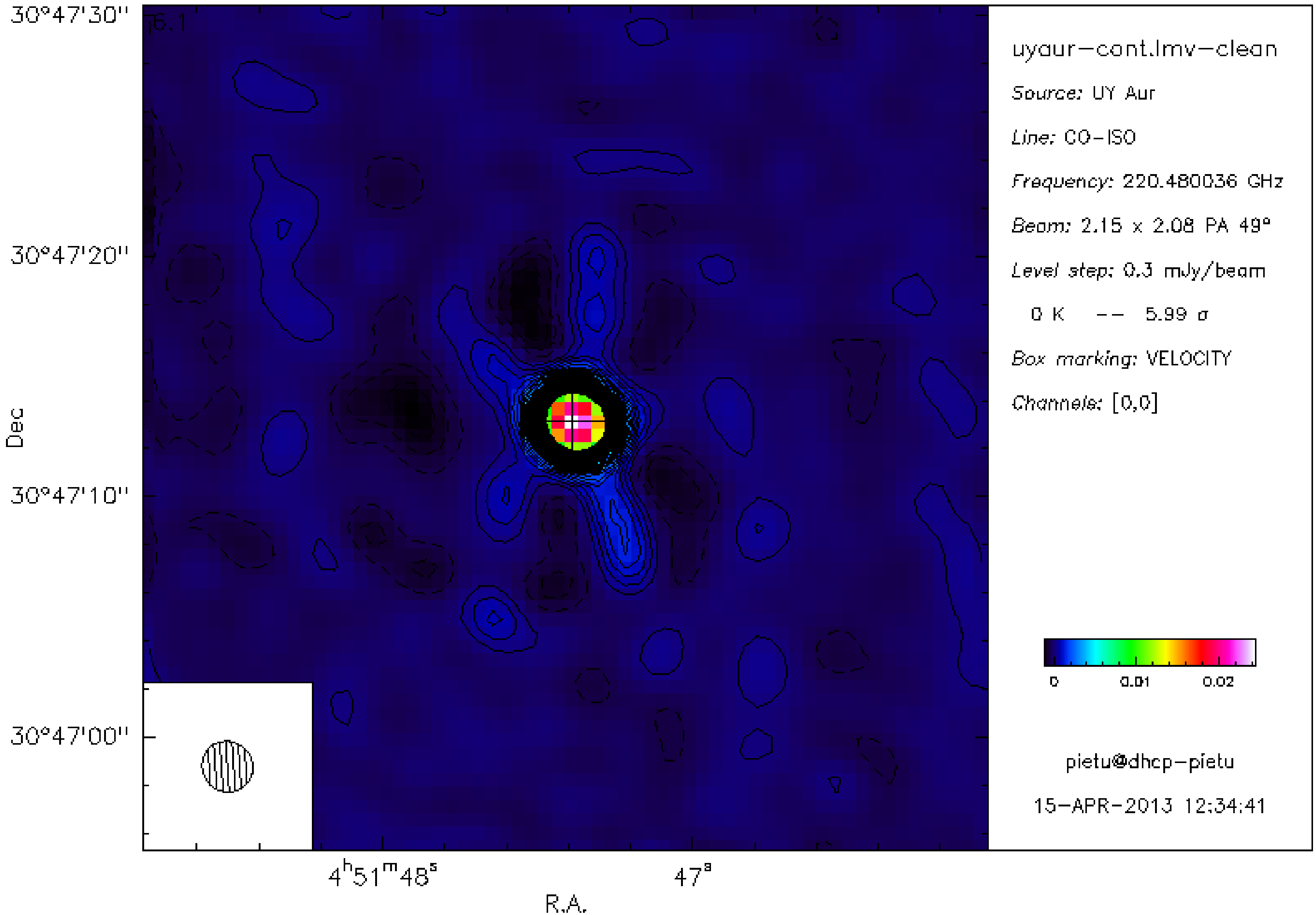
- For NOEMA (12 antennas, 32 GHz), this translates to

Band	1	2	3	4
Tsys	100	150	250	500
T=45 s	0.4 mJy	0.8 mJy	1.5 mJy	4 mJy
T=120 s	0.2 mJy	0.5 mJy	1 mJy	2.5 mJy

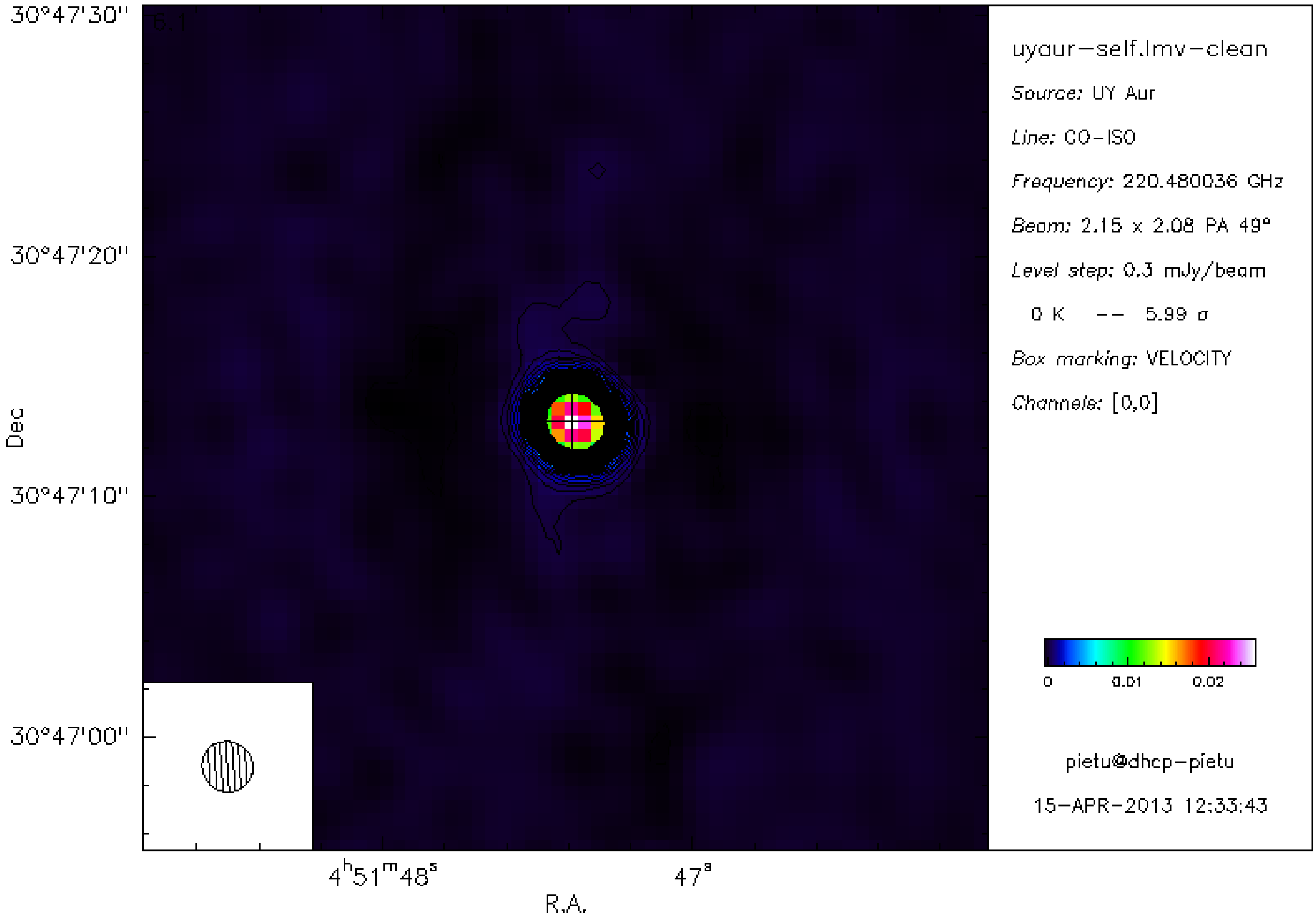
- Achtung ! Point source sensitivity. Actual numbers depend on your source structure.

Now the magic: it works !

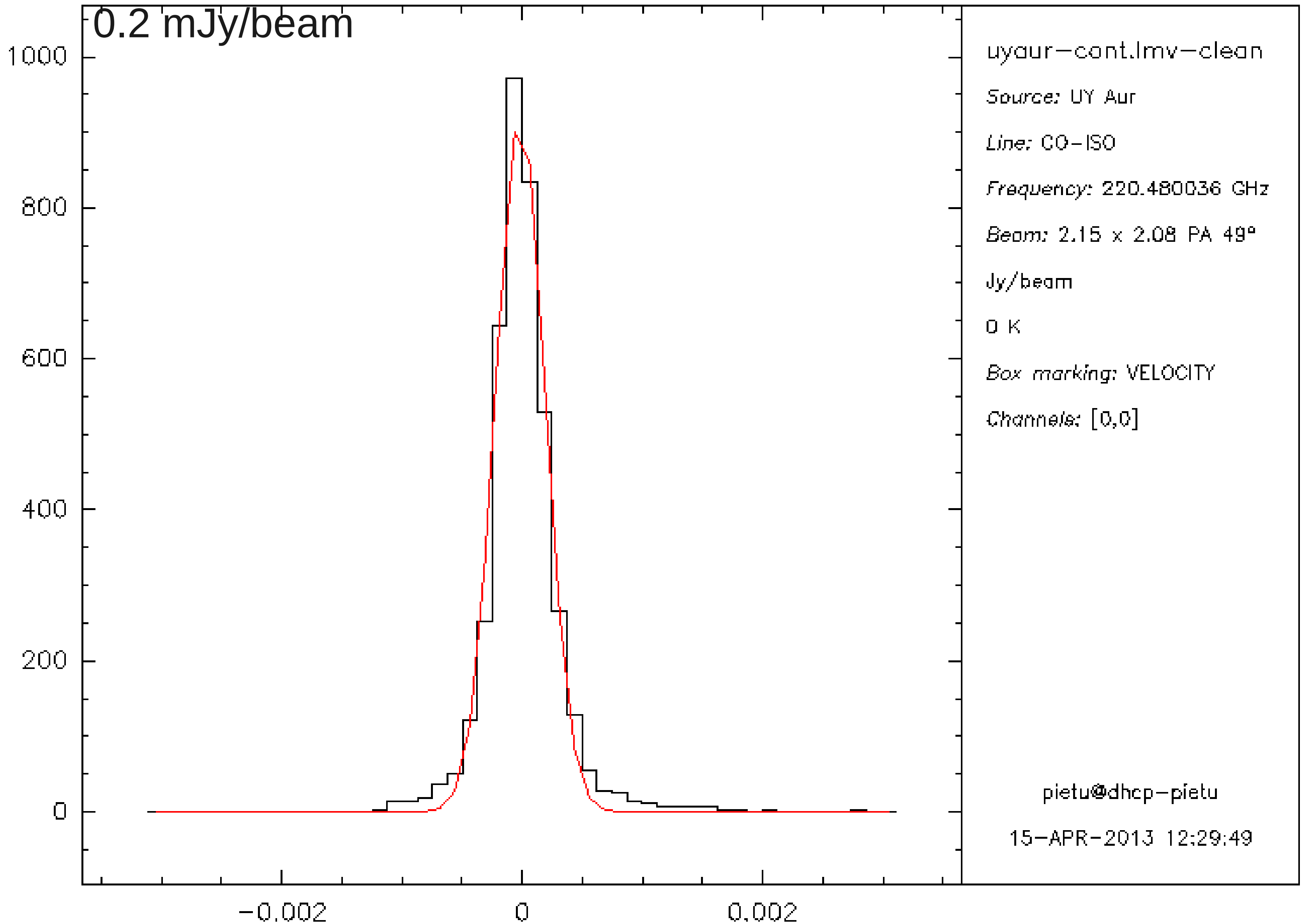
-



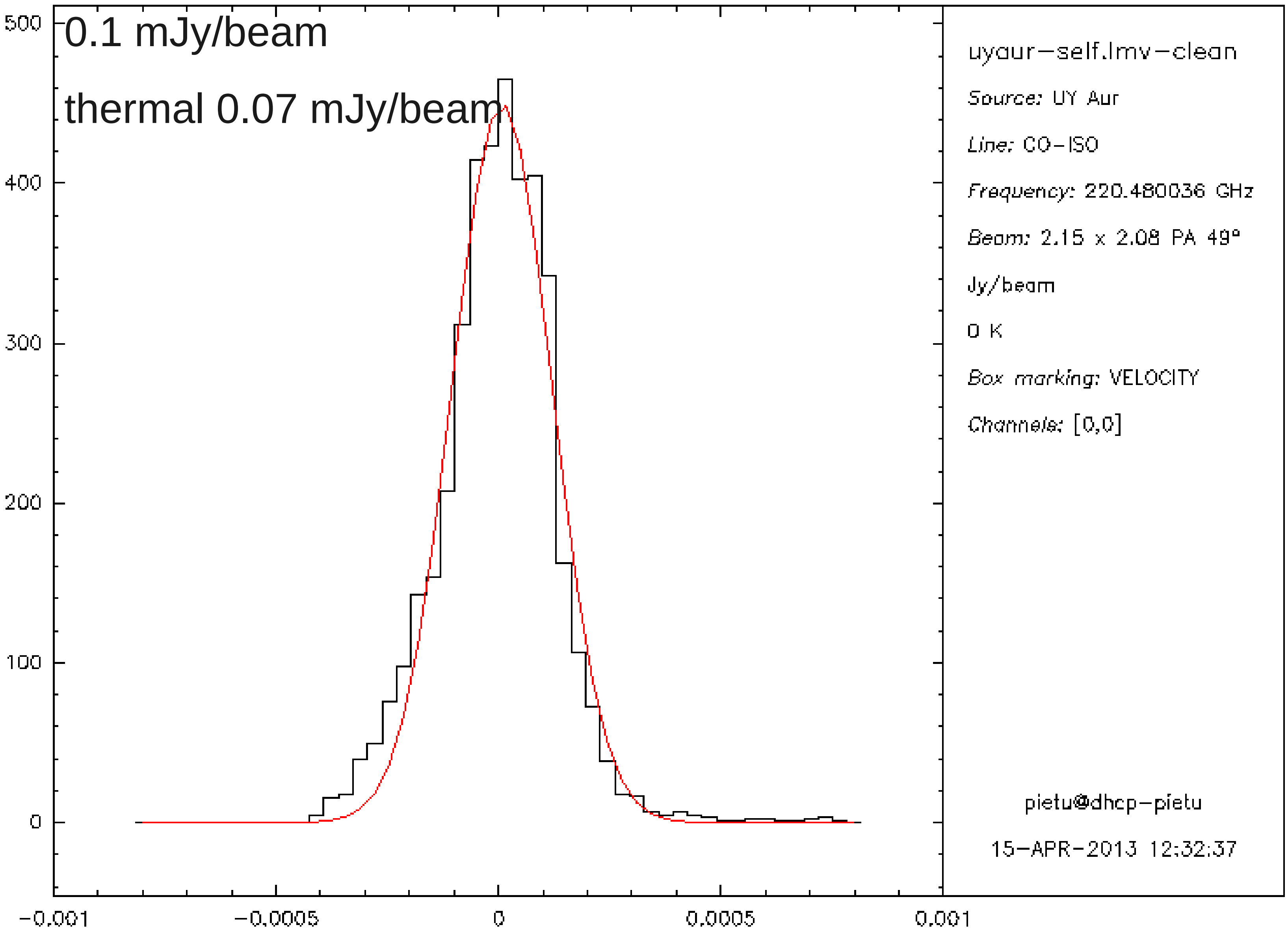
Now the magic: it works !



Now the magic: it works !

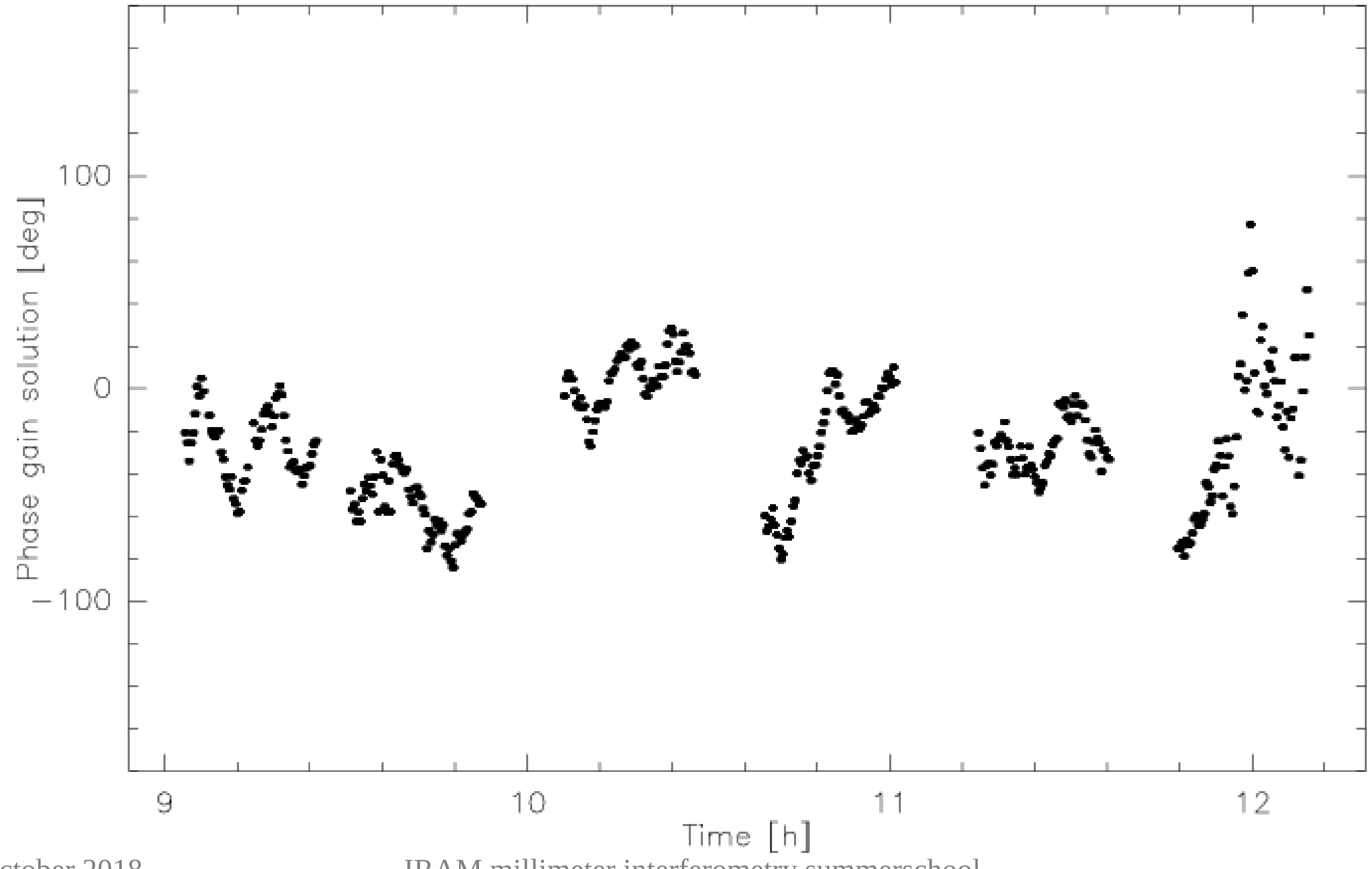


Now the magic: it works !

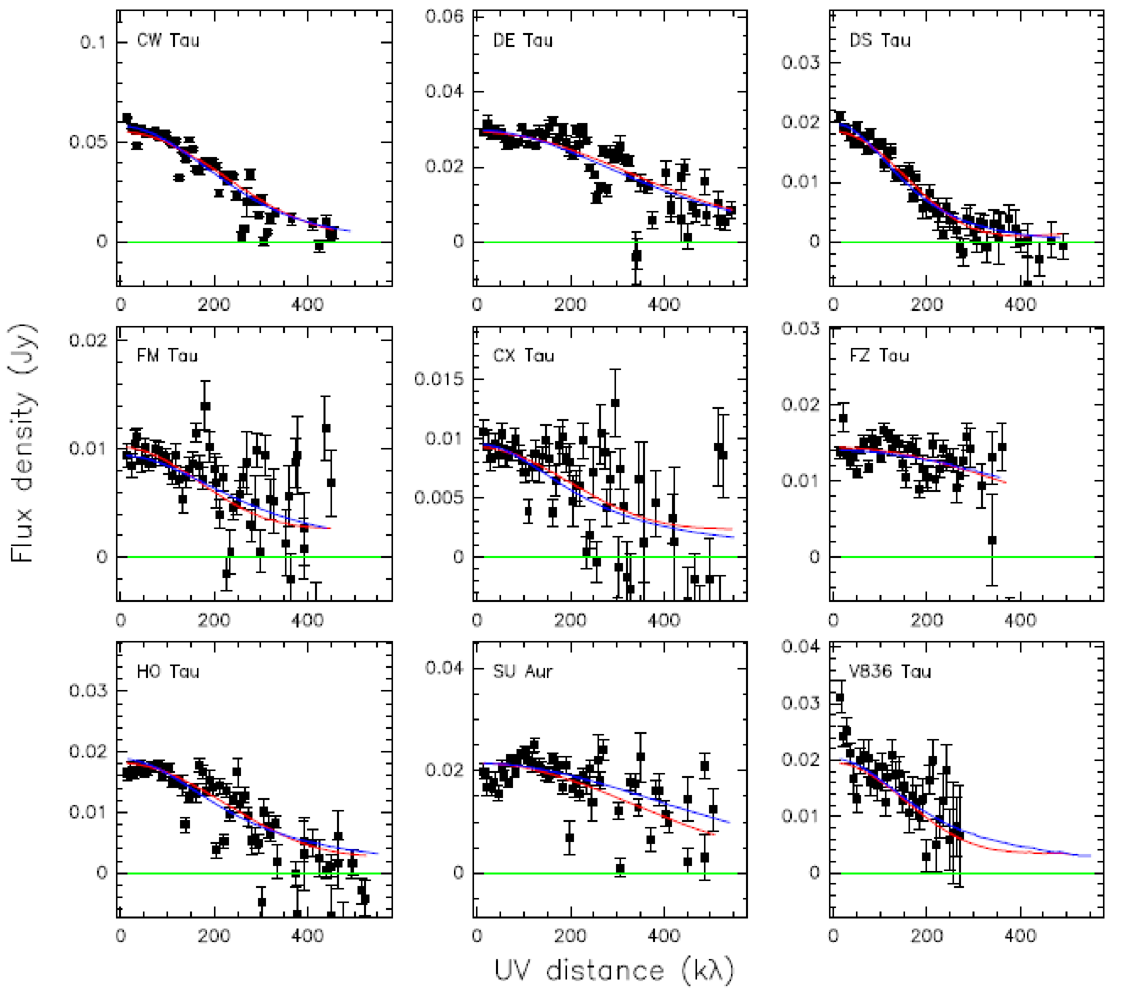


Now the magic

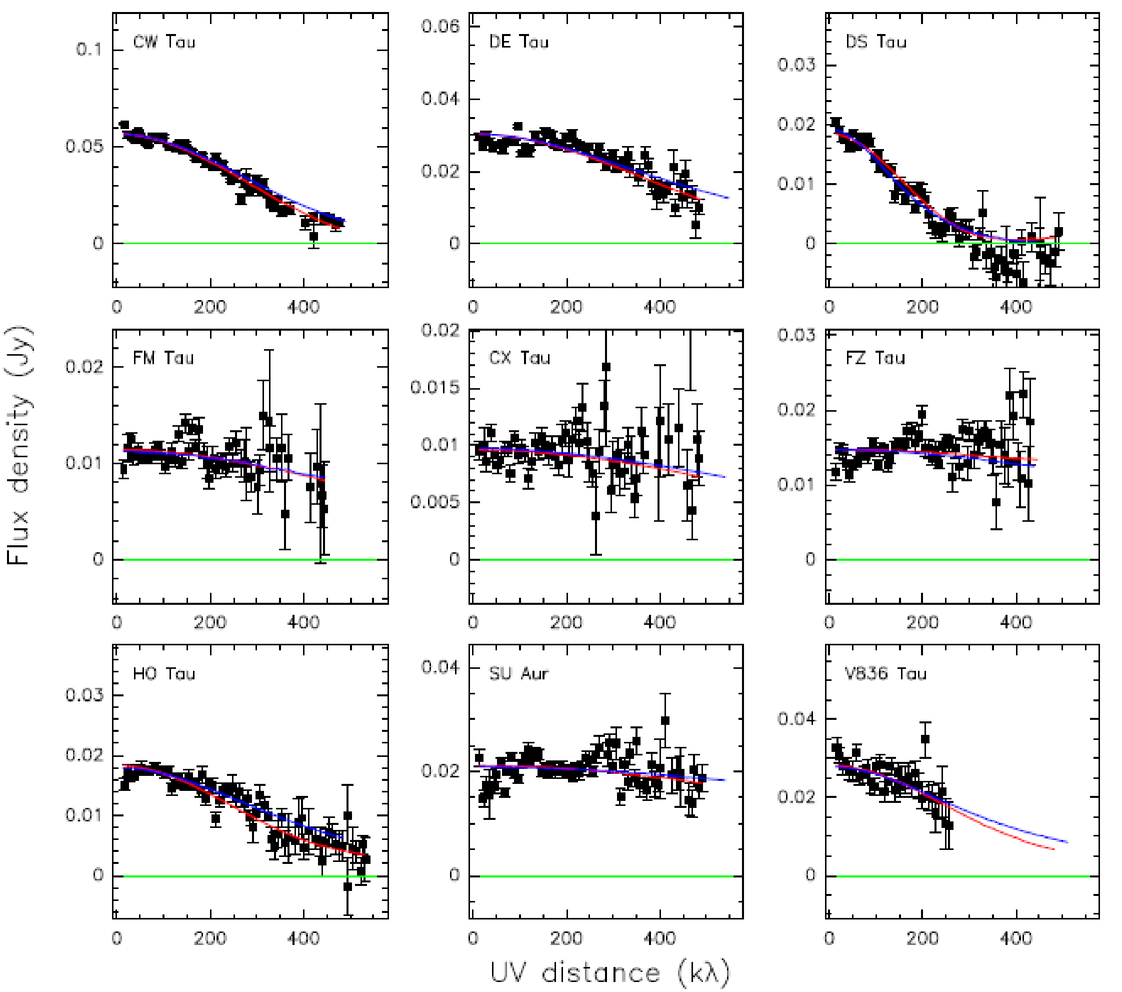
- Example of phase gain on one baseline

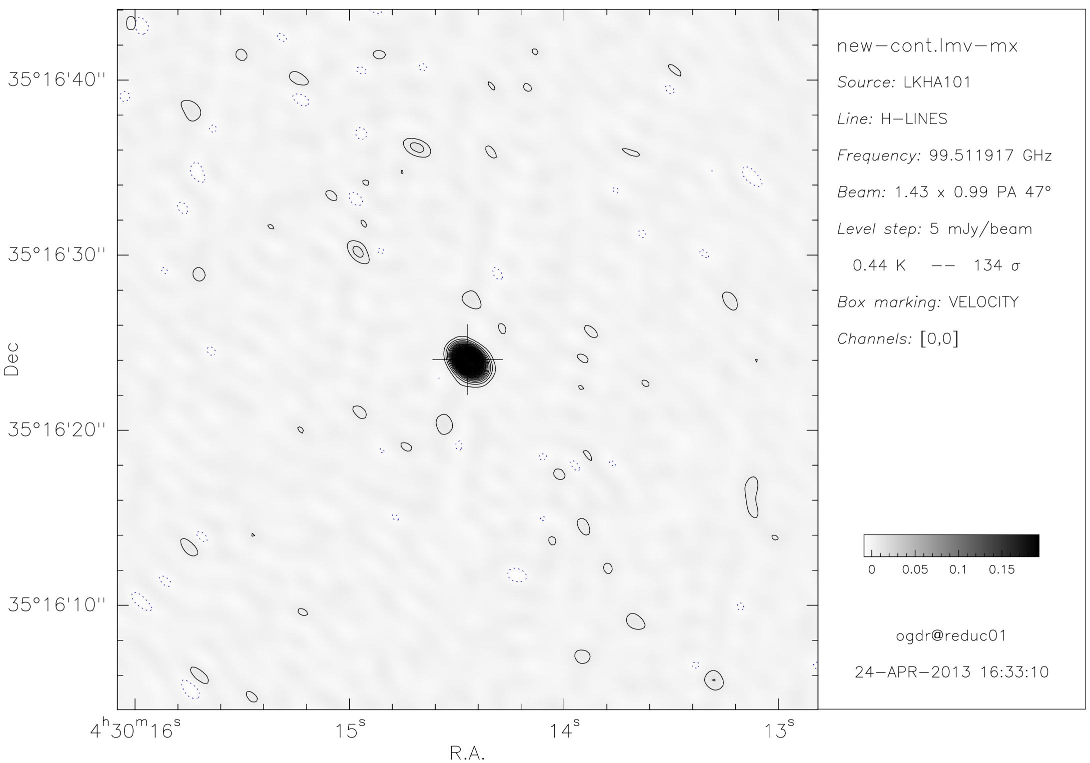


Results: survey of faint disks

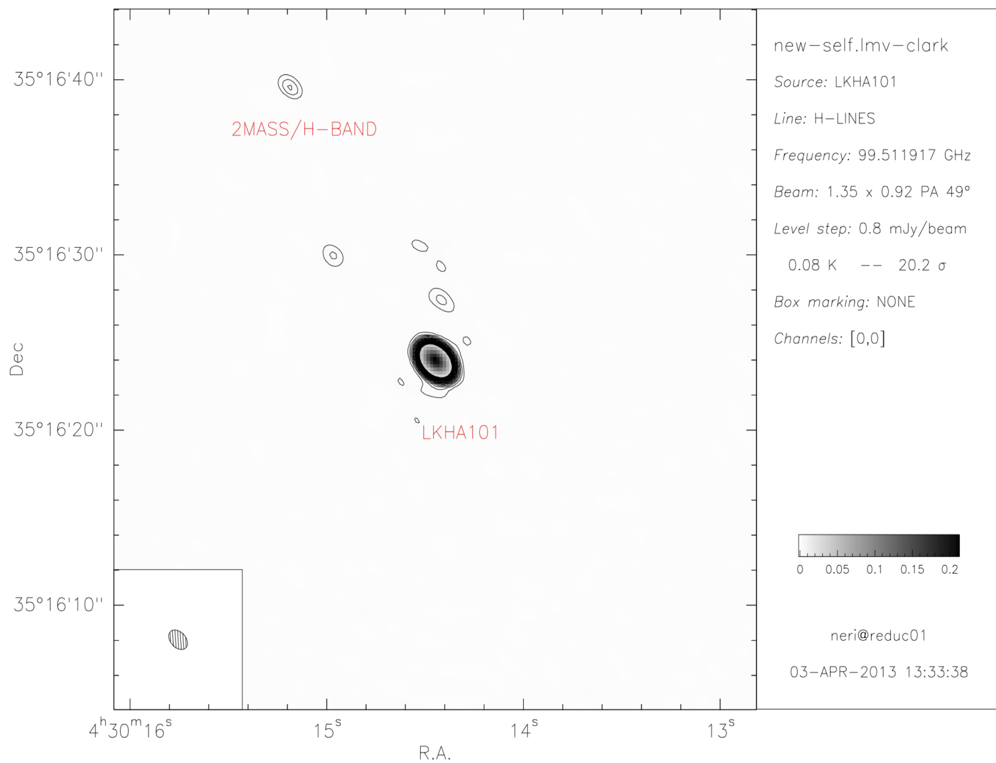


Results: survey of faint disks

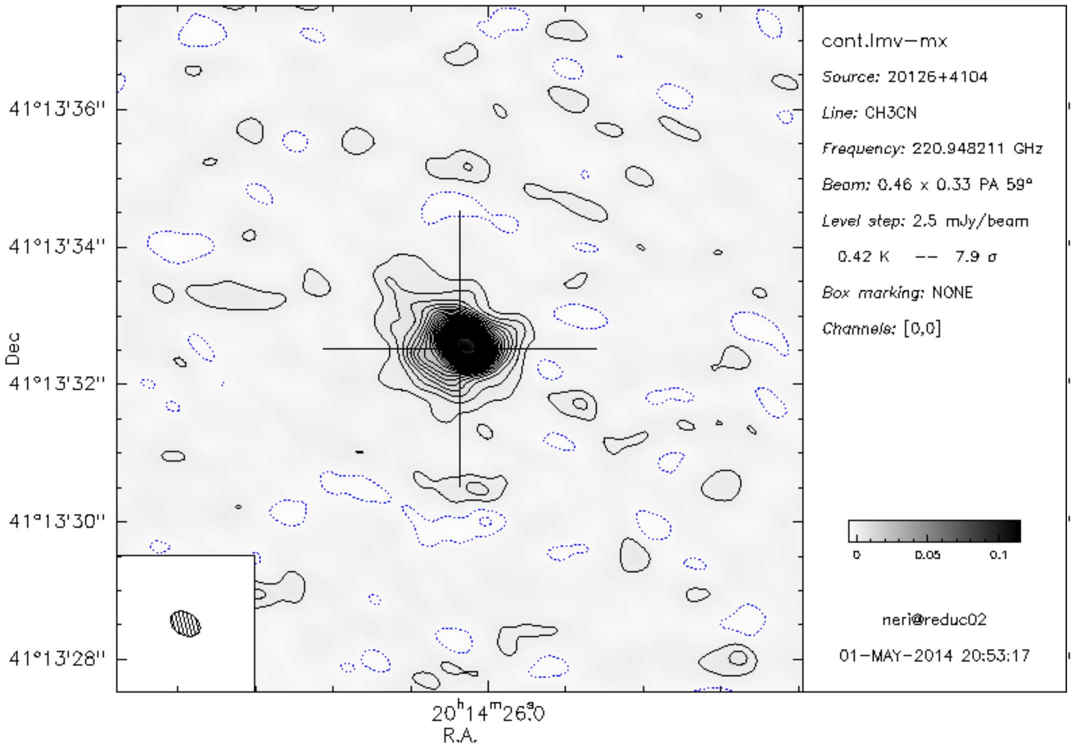


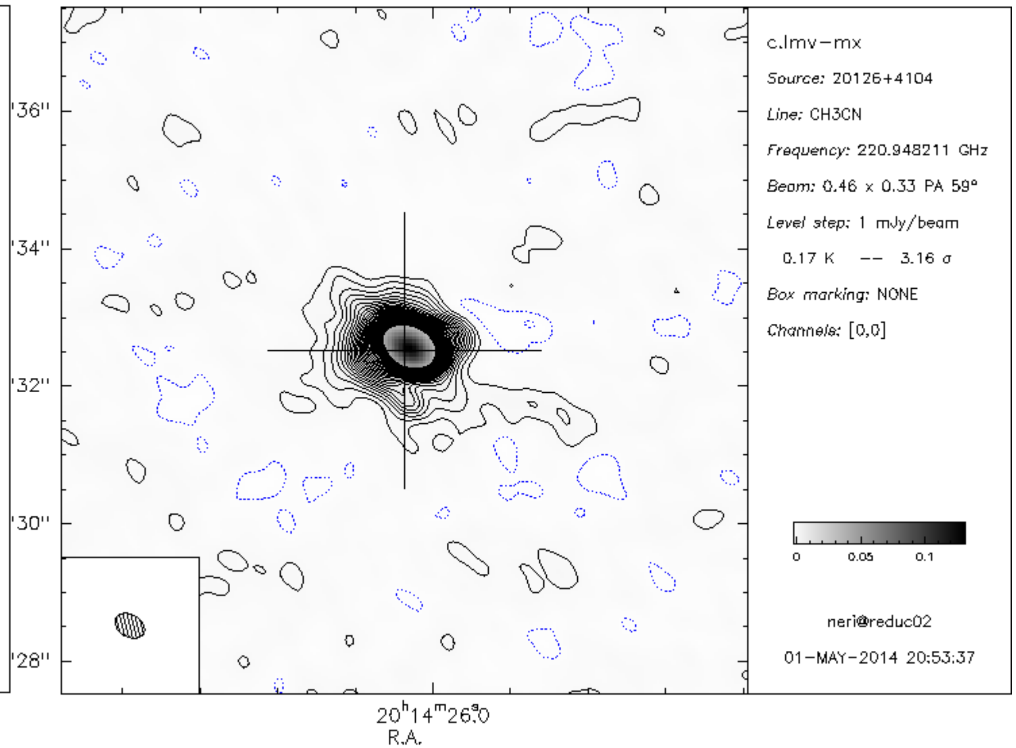


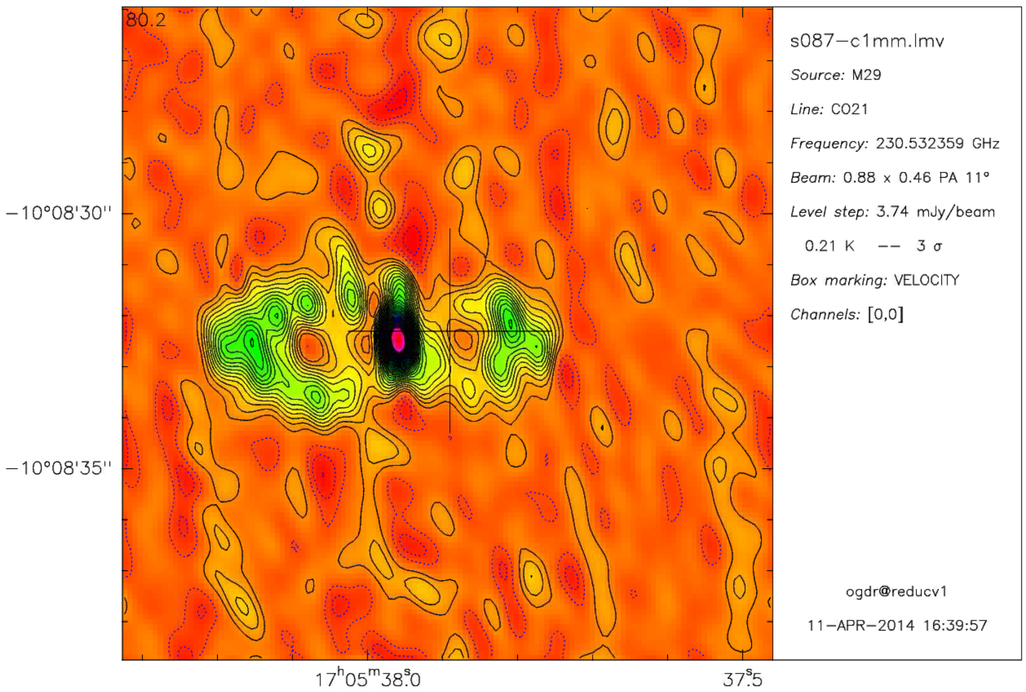
0
35°16'40''
35°16'30''
35°16'20''
35°16'10''
Dec
4h30m16s
15s
14s
13s
R.A.
new−cont.lmv−mx
Source: LKHA101
Line: H−LINES
Frequency: 99.511917 GHz
Beam: 1.43 x 0.99 PA 47°
Level step: 5 mJy/beam
0.44 K −− 134 σ
Box marking: VELOCITY
Channels: [0,0]
0 0.05 0.1 0.15
ogdr@reduc01
24−APR−2013 16:33:10

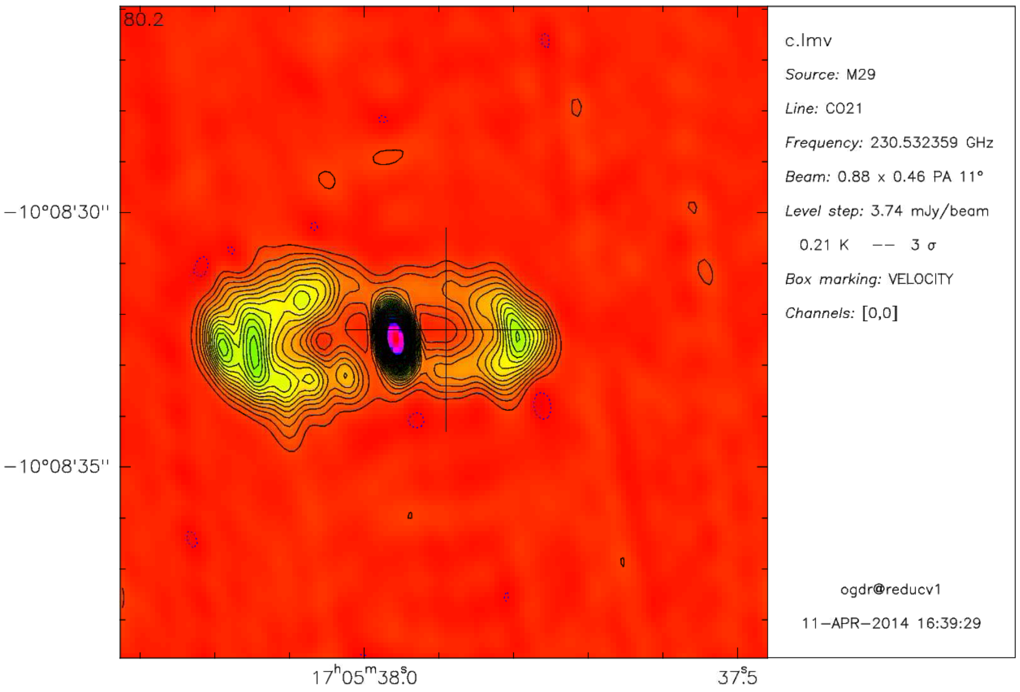


35°16'40''
35°16'30''
35°16'20''
35°16'10''
Dec
2MASS/H−BAND
LKHA101
4h30m16s
15s
14s
13s
R.A.
new−self.lmv−clark
Source: LKHA101
Line: H−LINES
Frequency: 99.511917 GHz
Beam: 1.35 x 0.92 PA 49°
Level step: 0.8 mJy/beam
0.08 K −− 20.2 σ
Box marking: NONE
Channels: [0,0]
0 0.05 0.1 0.15 0.2
neri@reduc01
03−APR−2013 13:33:38


cont.lmv−mx
Source: 20126+4104
Line: CH3CN
Frequency: 220.948211 GHz
Beam: 0.46 x 0.33 PA 59°
Level step: 2.5 mJy/beam
0.42 K −− 7.9 σ
Box marking: NONE
Channels: [0,0]
0 0.05 0.1
neri@reduc02
01−MAY−2014 20:53:17
41°13'36''
41°13'34''
41°13'32''
41°13'30''
41°13'28''
Dec
20h14m26s.0
R.A.


c.lmv−mx
Source: 20126+4104
Line: CH3CN
Frequency: 220.948211 GHz
Beam: 0.46 x 0.33 PA 59°
Level step: 1 mJy/beam
0.17 K −− 3.16 σ
Box marking: NONE
Channels: [0,0]
0 0.05 0.1
neri@reduc02
01−MAY−2014 20:53:37
'36''
'34''
'32''
'30''
'28''
20h14m26s.0
R.A.


80.2
s087−c1mm.lmv
Source: M29
Line: CO21
Frequency: 230.532359 GHz
Beam: 0.88 x 0.46 PA 11°
Level step: 3.74 mJy/beam
0.21 K −− 3 σ
Box marking: VELOCITY
Channels: [0,0]
ogdr@reducv1
11−APR−2014 16:39:57
−10°08'30''
−10°08'35''
17h05m38s.0
37s.5

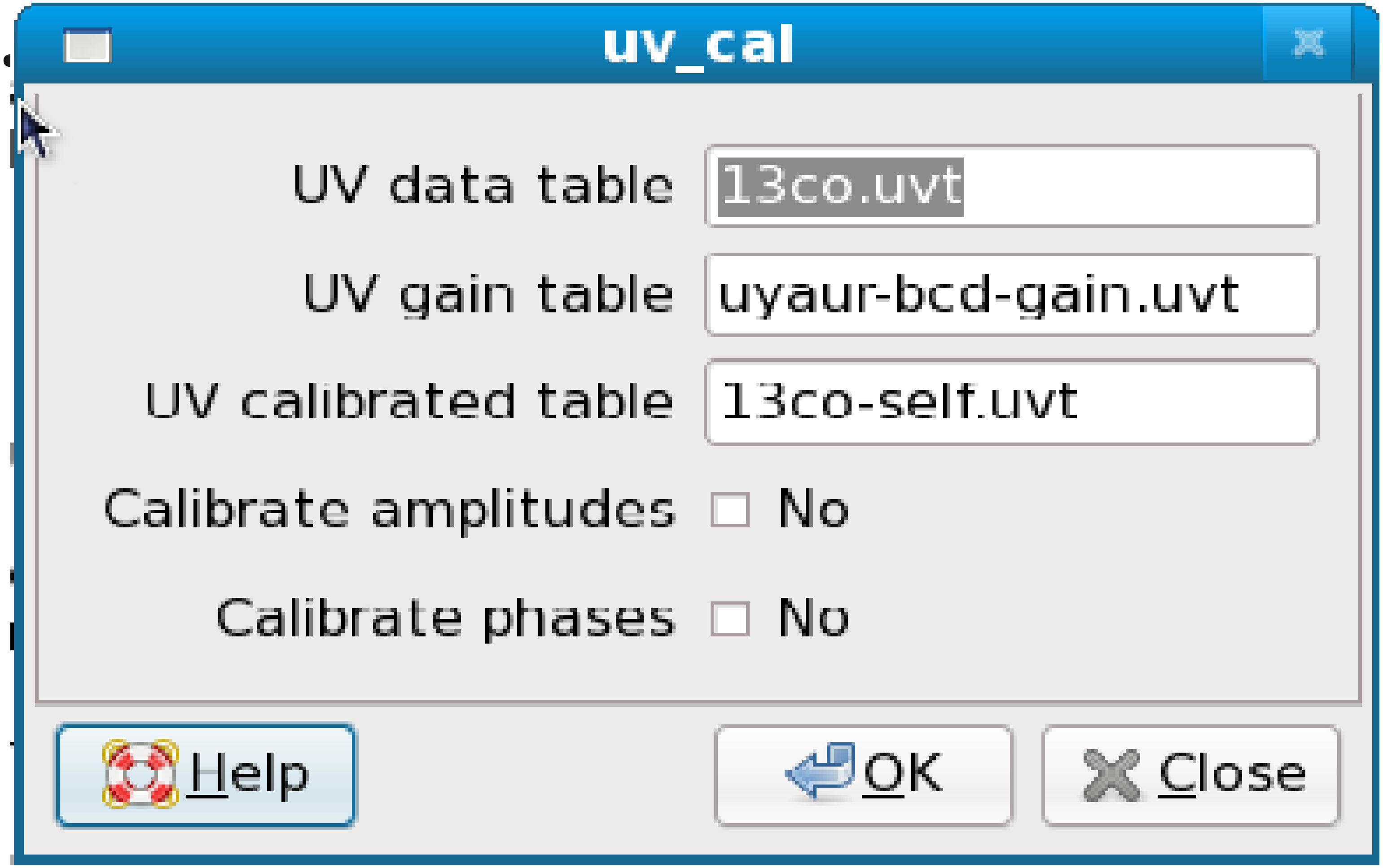

80.2
c.lmv
Source: M29
Line: CO21
Frequency: 230.532359 GHz
Beam: 0.88 x 0.46 PA 11°
Level step: 3.74 mJy/beam
0.21 K −− 3 σ
Box marking: VELOCITY
Channels: [0,0]
ogdr@reducv1
11−APR−2014 16:39:29
−10°08'30''
−10°08'35''
17h05m38s.0
37s.5

This solution can be transferred

- A model was derived for a single channel (a given channel from a line cube or continuum).
- Possibility to use the gain table derived to correct a whole data cube (i.e. to use the gain derived from the continuum to correct the line data)
- Run uv_cal task

This solution can be transferred

- A model was derived for a single channel (a given channel from a line cube or continuum).
- Possibility to use the gain table derived to correct a whole data cube (i.e. to use the gain derived from the continuum to correct the line data)



Conclusions

- Working version of selfcalibration
- Shown to improve situation w/o major artefact in many cases.
- Only the phase selfcalibration tested.
- Be careful if you plan to use the amplitude selfcalibration (non-conservation of the fluxes).
- Try out by yourself, but be conservative and critical. Especially, you could create a source where there is none.
- Allows high-dynamic range imaging for NOEMA.
- There is no limit to what you can do. You can implement your own version of self-calibration. Caveat, you will bias the image toward your initial model.