Imaging, Deconvolution & Image Analysis
II. Practice

Jérôme PETY
(IRAM/Obs. de Paris)

6th IRAM Millimeter Interferometry School
Oct. 6 - Oct. 10 2008, Grenoble

From Calibrated Visibilities to Images: Summary

Fourier Transform and Deconvolution:
The two key issues in imaging.

Stage	Implementation
Calibrated Visibilities	
⇓ Fourier Transform	`GO UVSTAT, GO UVMAP`
Dirty beam & image	
⇓ Deconvolution	`GO CLEAN`
Clean beam & image	
⇓ Visualization	`GO BIT, GO VIEW`
⇓ Image analysis	`GO NOISE, GO FLUX, GO MOMENTS`
Physical information on your source	

Weighting and Tapering: GILDAS implementation

Resolution, point/extended source sensitivity
as a function of
robust threshold or tapering distance.

```
MAPPING> GO UVSTAT WEIGHT

Robust      Major    Minor     PA              Noise      Sidelobe
              (")      (")     (deg)    (mJy)     (K)       %
   0.10      0.840    0.527     18.5     0.791    0.044  -16.1   13.9
  None       1.271    0.940    155.1     0.406    0.008   -2.7   25.9
I-UV_STAT,  Recommended pixel size is  0.080 -  0.264"

MAPPING> GO UVSTAT TAPER

Taper       Major    Minor     PA              Noise      Sidelobe
 (m)          (")      (")     (deg)    (mJy)     (K)       %
  30.00      4.844    3.817     11.9     1.019    0.001  -44.7   14.5
  None       1.323    0.967    151.5     0.406    0.008  -12.9   28.7
```

From Calibrated Visibilities to Images: GILDAS implementation

`MAPPING> INPUT UVMAP`: display `GO UVMAP` control parameters.
Default = 0 ⇒ Try something clever.

- Gridding kernel: `CONVOLUTION` (default: 5, *i.e.* spheroidals).
- Data cube setup:
  - First and last data channels to map: `MCOL` (default: 0, *i.e.* all channels).
  - Pixel size: `MAP_CELL` (in arcsec, default: 0, *i.e.* 1/3 to 1/4 synthesized beam).
  - Map size: `MAP_SIZE` (in pixels, default: 0, *i.e.* $\sim$ twice primary beam size).
- Weighting:
  - kind: `WEIGHT_MODE`
    (Possibilities: `NATURAL/UN`. Default: `NATURAL`).
  - Size of uv plane cell and robust threshold: `UV_CELL`
    (Defaults: 7.5 m, 1).
- Tapering: Gaussian taper parameters `UV_TAPER` (in m and degree, default: 0, *i.e.* no tapering).

From Calibrated Visibilities to Images: Summary

Fourier Transform and Deconvolution:
The two key issues in imaging.

Stage	Implementation
Calibrated Visibilities	
⇓ Fourier Transform	`GO UVSTAT, GO UVMAP`
Dirty beam & image	
⇓ Deconvolution	`GO CLEAN`
Clean beam & image	
⇓ Visualization	`GO BIT, GO VIEW`
⇓ Image analysis	`GO NOISE, GO FLUX, GO MOMENTS`
Physical information on your source	

Deconvolution: GILDAS Implementation

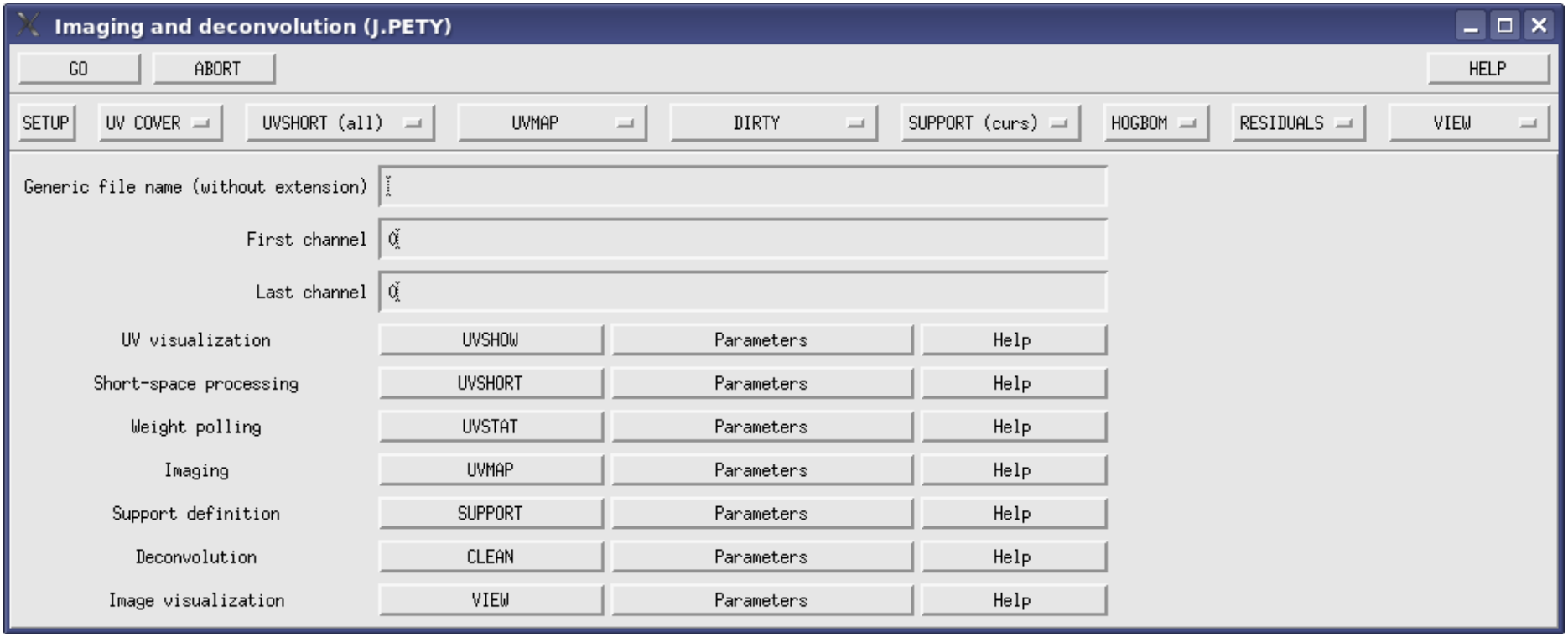
1. MAPPING Command Line

`MAPPING> INPUT CLEAN`: display CLEAN control parameters.

- Loop gain: `GAIN` (default: 0.2).
- Stopping criterions:
  - Maximum number of iterations: `NITER` (default: 100).
  - Maximum residual amplitude, expressed as fraction of peak intensity: `FRES` (default: 0.05).
  - Maximum residual amplitude, in map units (Jy/Beam): `ARES` (default: 0).
- Support: `BLC` and `TRC` are the Bottom Left Corner and Top Right Corner of a square support in pixel units (default: 0, *i.e.* no support).
- Gaussian clean beam parameters: `MAJOR, MINOR, ANGLE` (Default: 0, *i.e.* Fit dirty beam).

Deconvolution: GILDAS Implementation

2. MAPPING Window Interface



Typical MAPPING session

Look at the mapping-tutorial.map procedure for details.

```
MAPPING> let name 1mm
MAPPING> go image

MAPPING> let name 1mm
MAPPING> go uvcov
MAPPING> go uvshow

MAPPING> go uvmap
MAPPING> go plot beam
MAPPING> go plot dirty

MAPPING> go clean
MAPPING> go plot cct
MAPPING> go plot res
MAPPING> go plot clean

MAPPING> exit
```

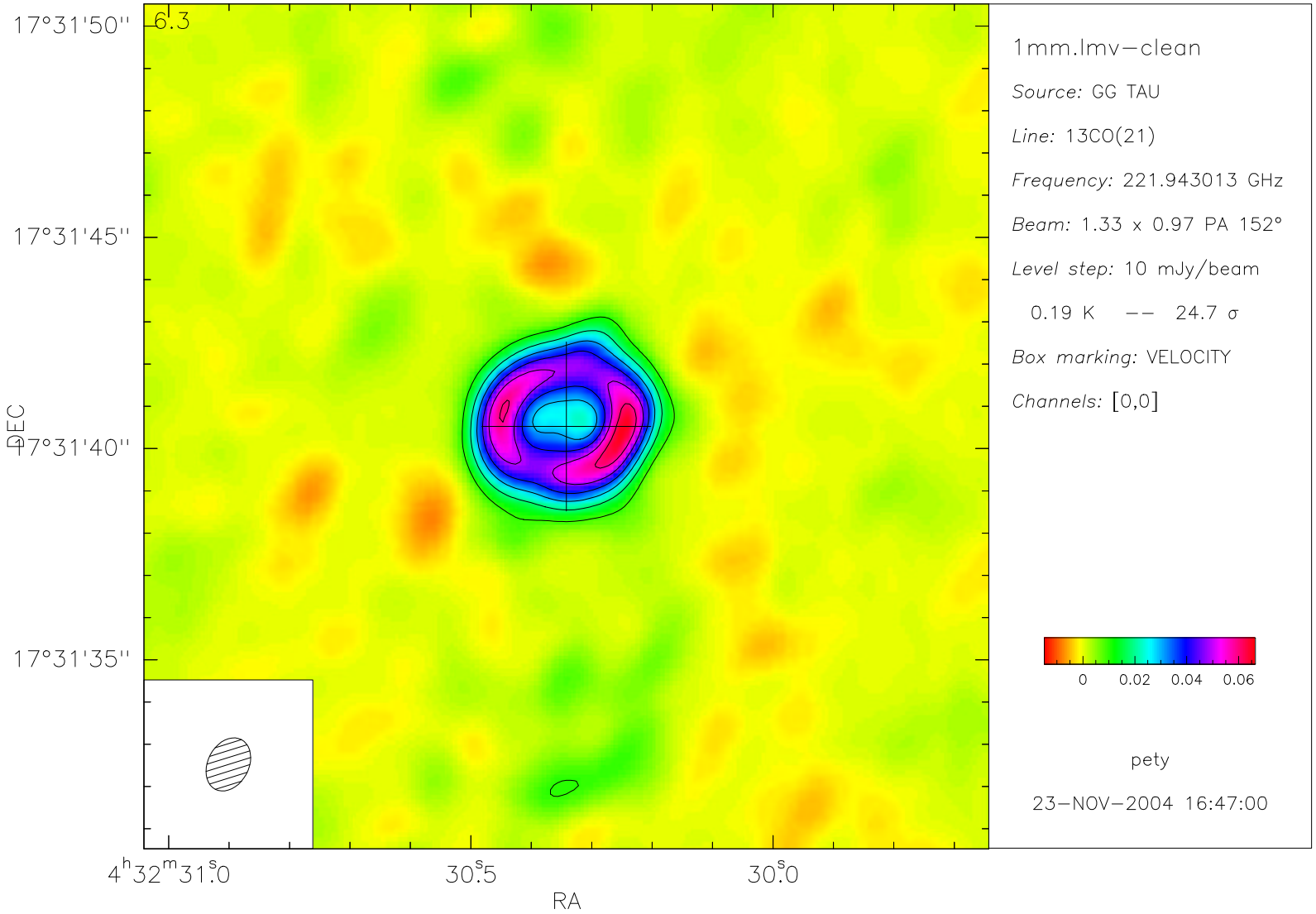
Visualization and Image Analysis

Fourier Transform and Deconvolution:
The two key issues in imaging.

Stage	Implementation
Calibrated Visibilities	
⇓ Fourier Transform	`GO UVSTAT, GO UVMAP`
Dirty beam & image	
⇓ Deconvolution	`GO CLEAN`
Clean beam & image	
⇓ Visualization	`GO BIT, GO VIEW`
⇓ Image analysis	`GO NOISE, GO FLUX, GO MOMENTS`
Physical information on your source	

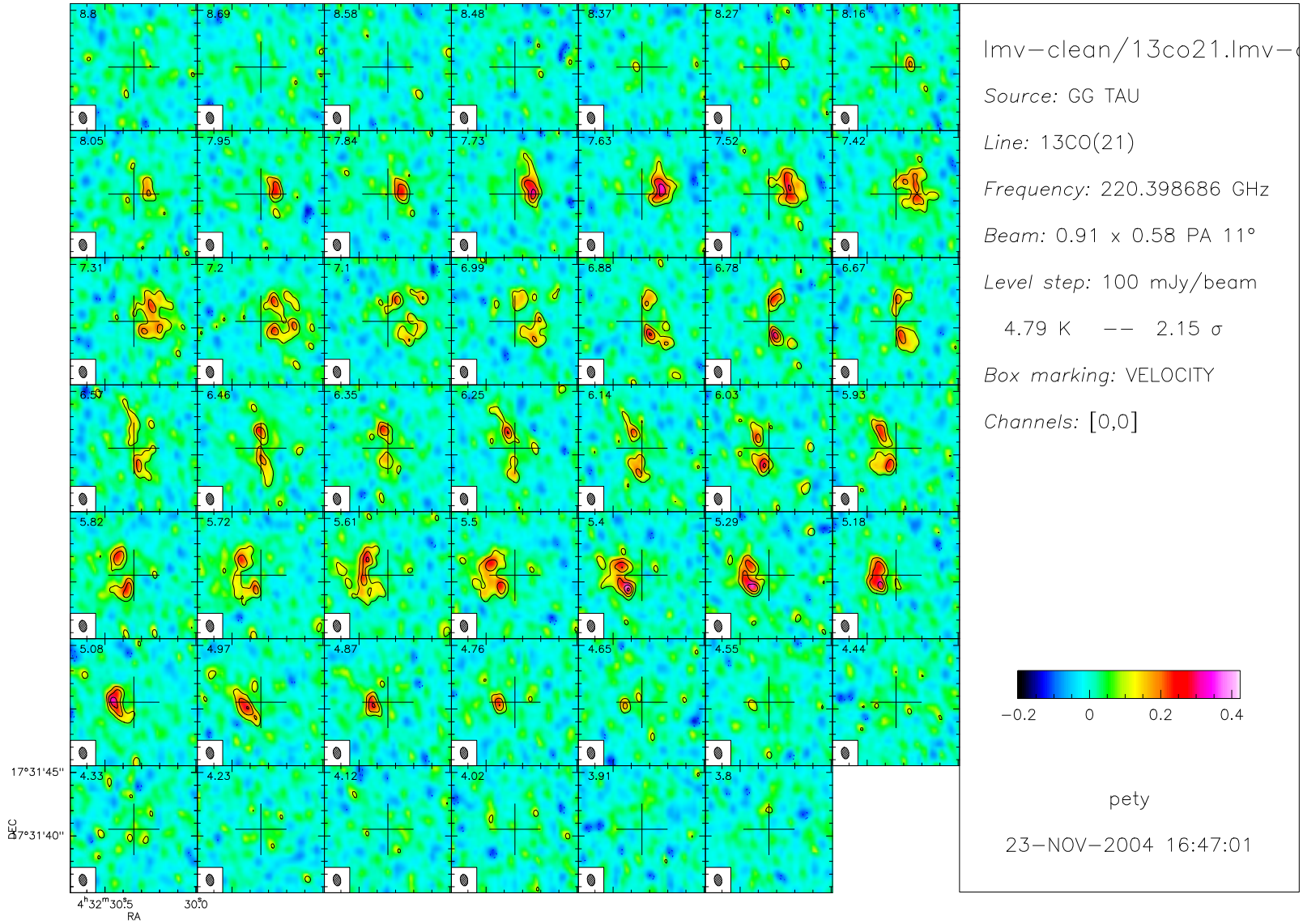
Visualization: I. GO BIT for continuum

```
MAPPING> let name 1mm
MAPPING> let type lmv-clean
MAPPING> go bit
MAPPING> input bit
```



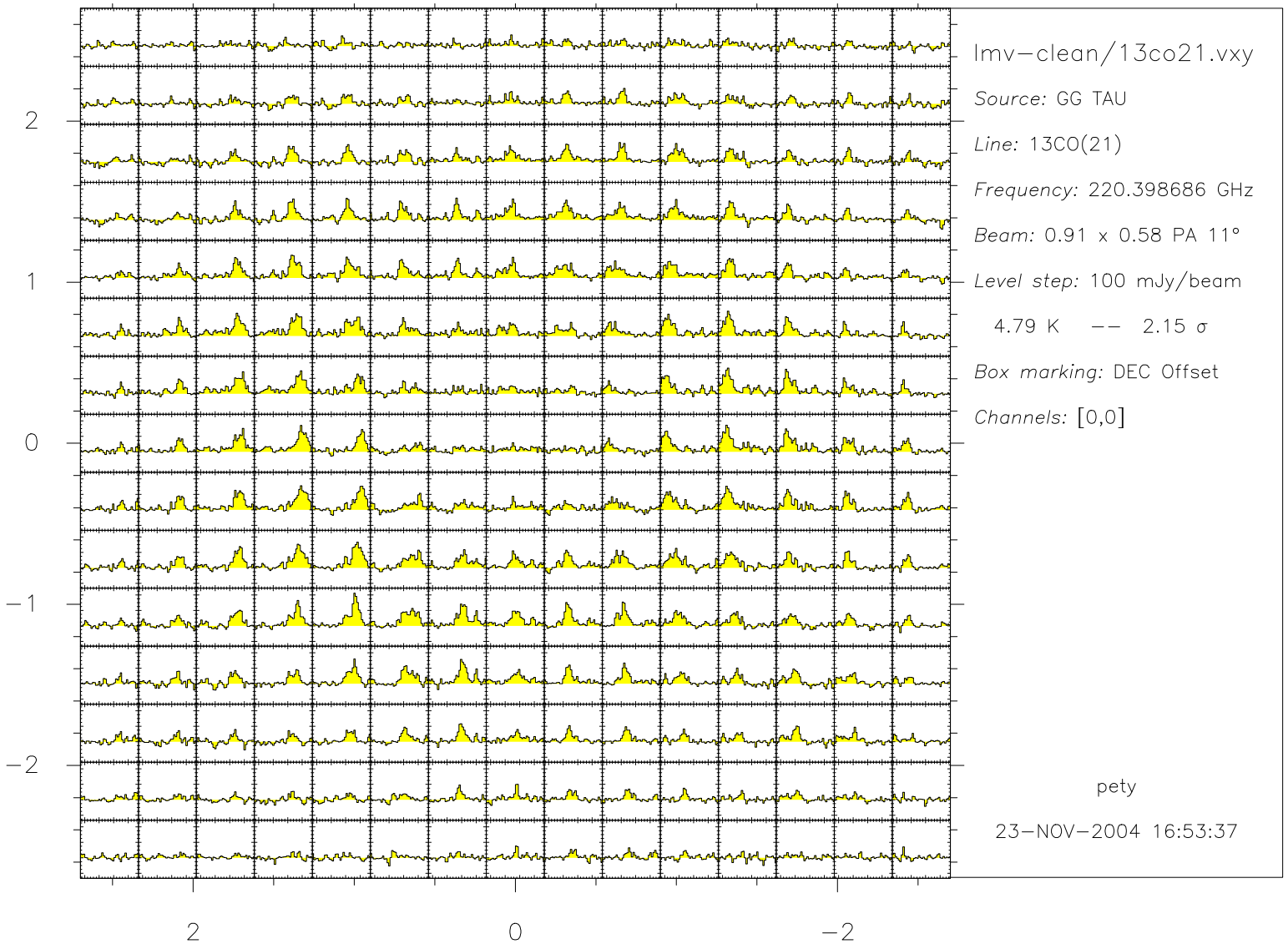
Visualization: II. GO BIT for spectra

```
MAPPING> let name 13co21
MAPPING> let type lmv-clean
MAPPING> go bit
```



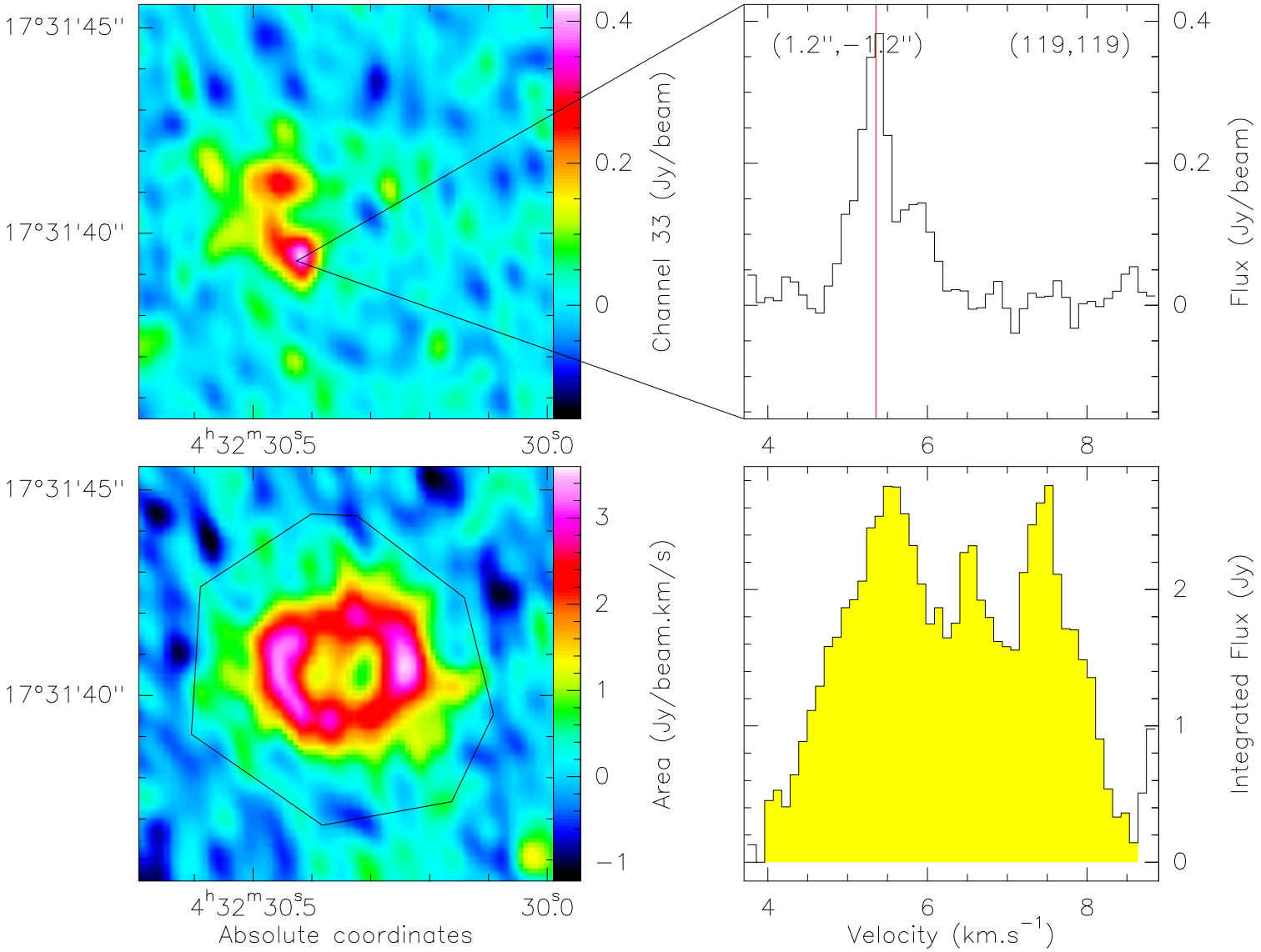
Visualization: III. GO SPECTRUM for spectra

```
MAPPING> let name 13co21
MAPPING> let type lmv-clean
MAPPING> go spectrum
```



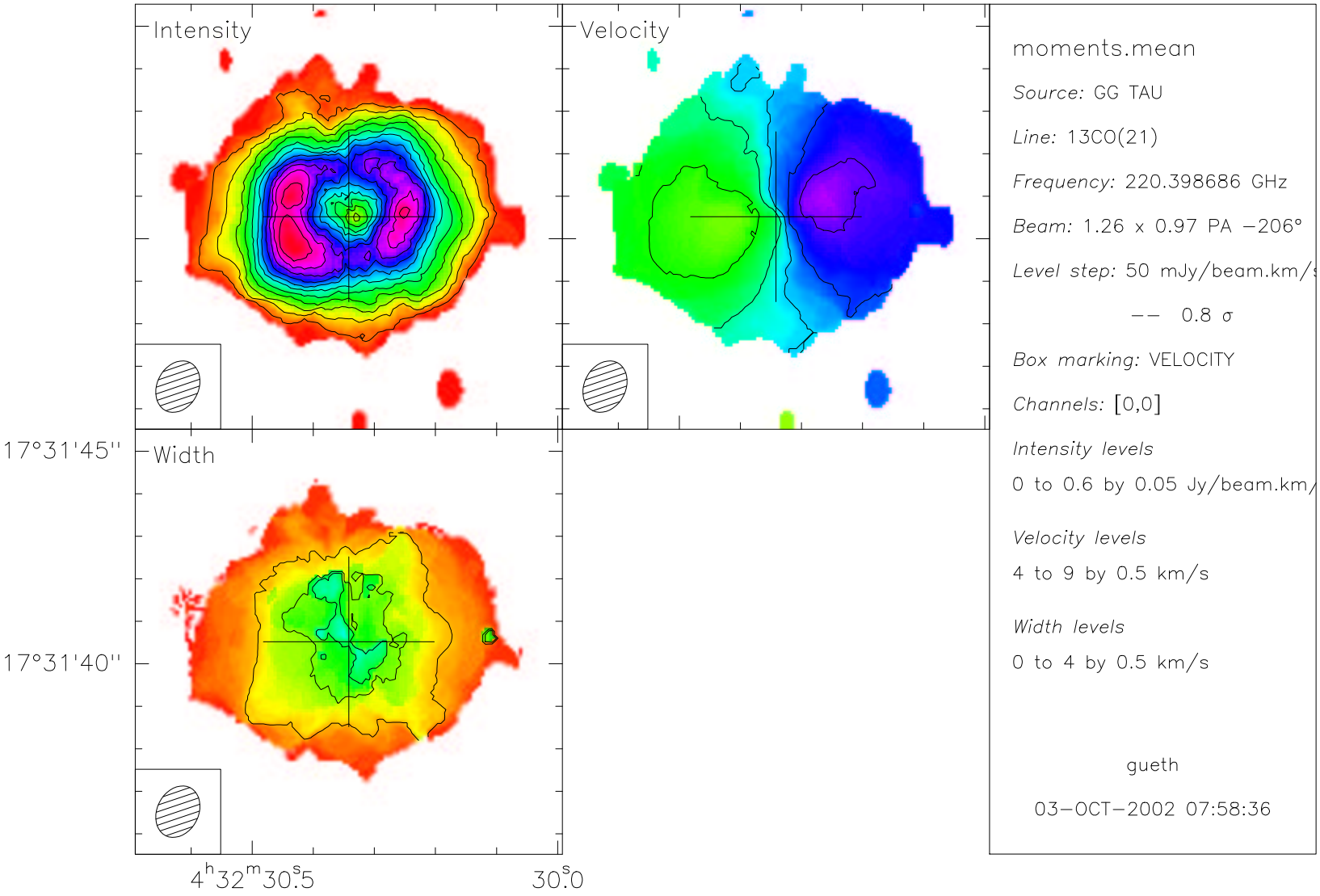
Visualization: IV. GO VIEW for spectra

```
MAPPING> let name 13co21
MAPPING> let type lmv-clean
MAPPING> go view
```



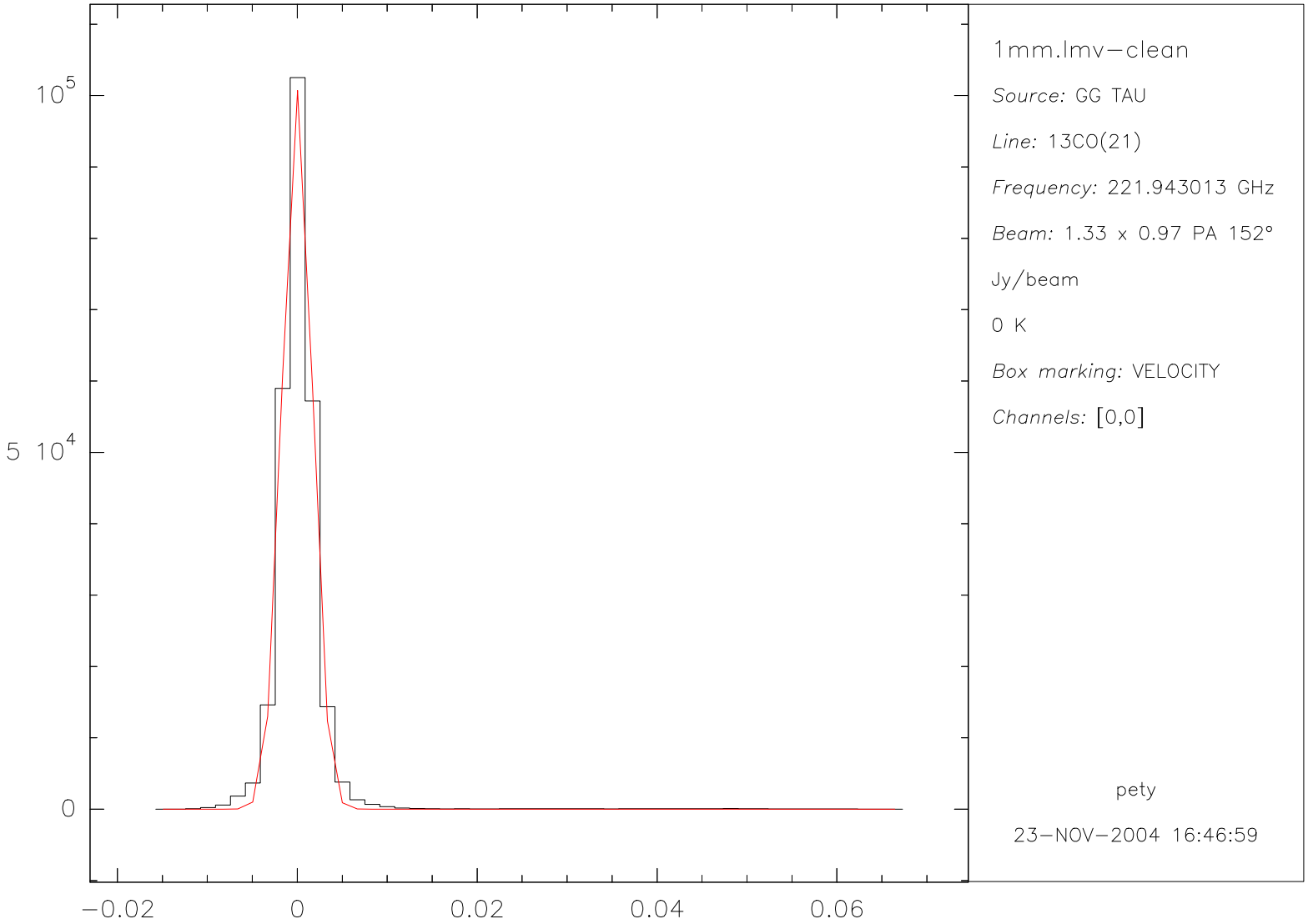
Analysis: I. Moment Estimation with GO MOMENT + GO VELOCITY

```
MAPPING> let name 13co21
MAPPING> let type lmv-clean
MAPPING> go moments
MAPPING> go velocity
```



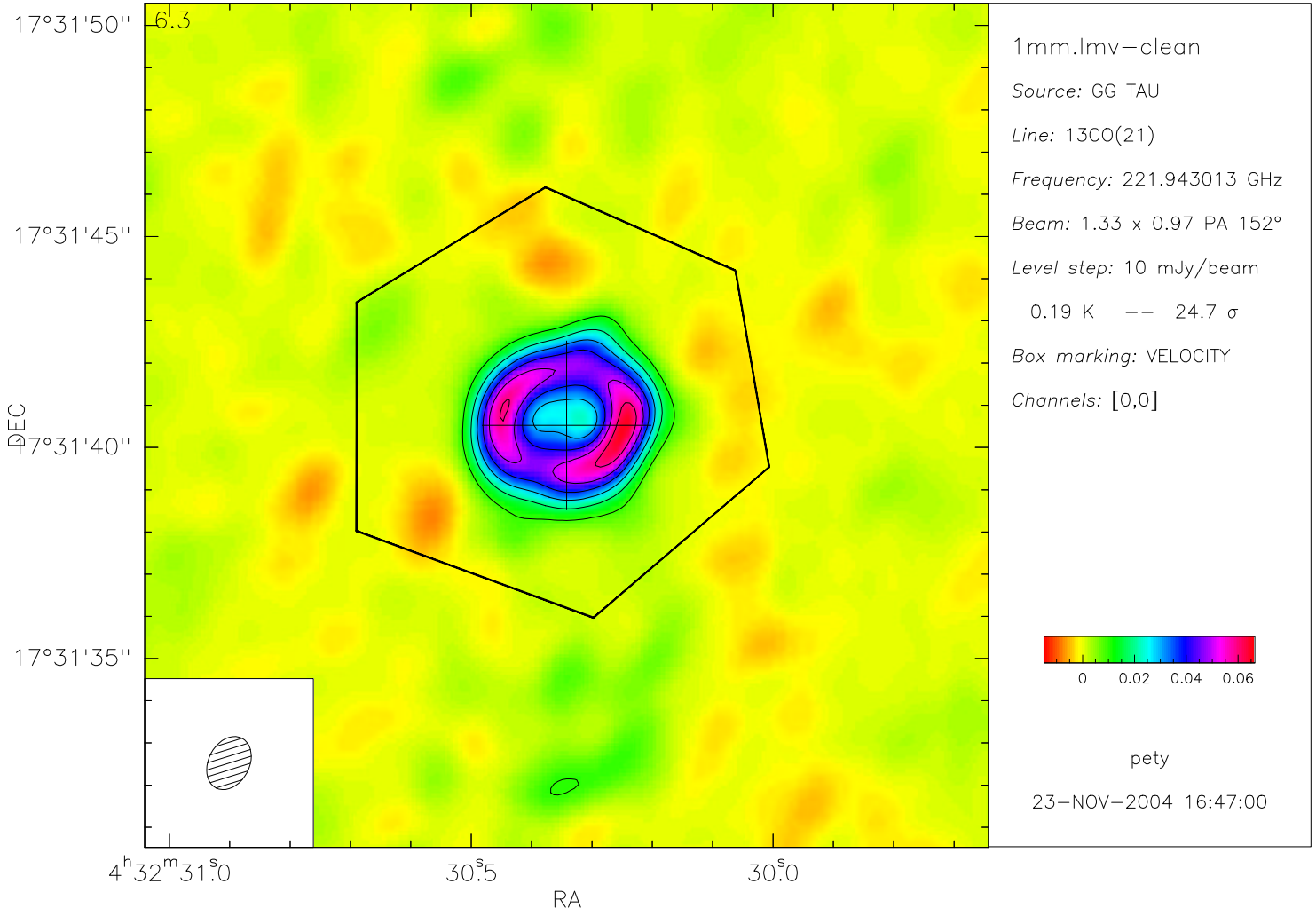
Analysis: II. Noise Estimation with GO NOISE

```
MAPPING> let name 1mm
MAPPING> let type lmv-clean
MAPPING> go noise
NOISE = 1.6312926E-03 !  Real GLOBAL
```

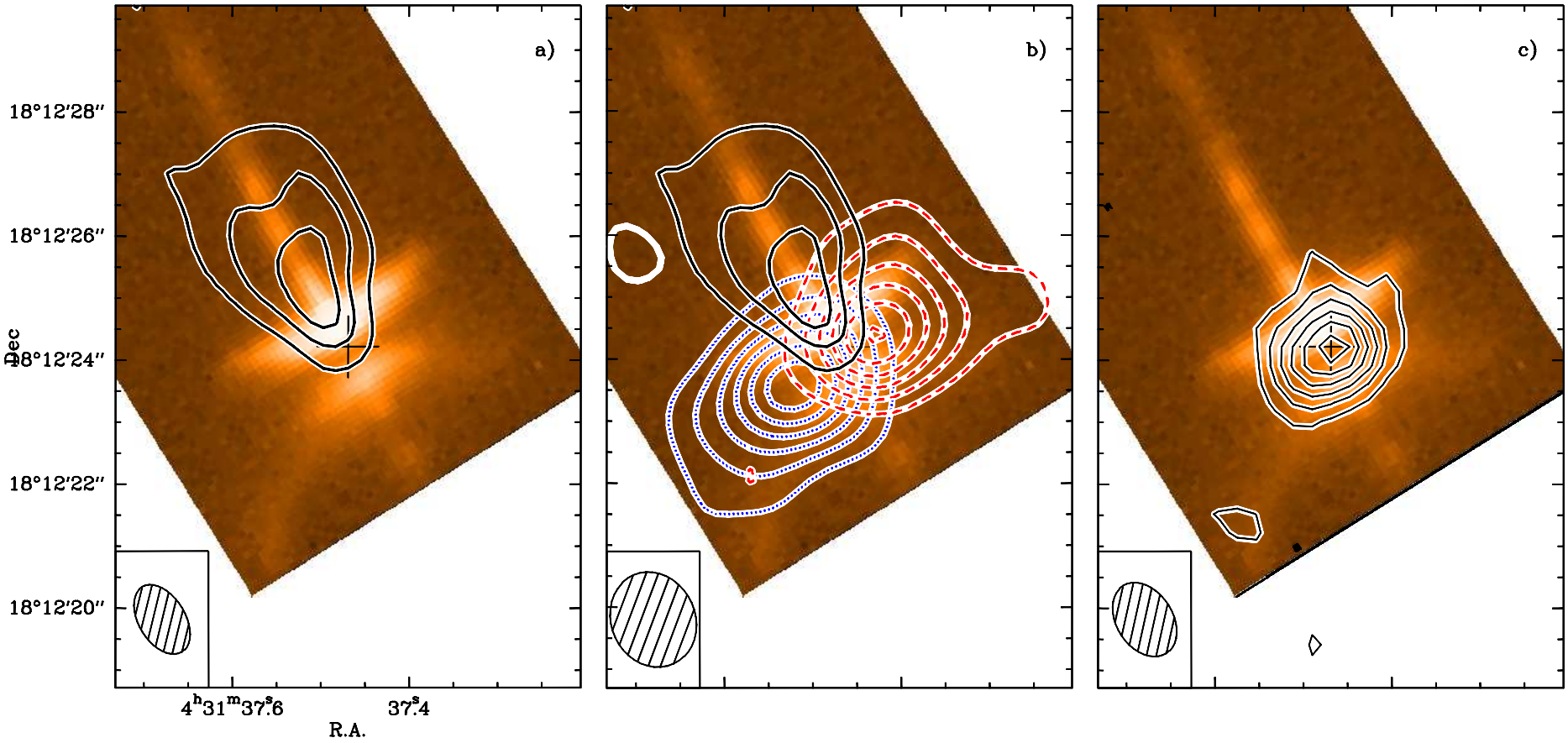


Analysis: III. Flux Estimation with GO FLUX

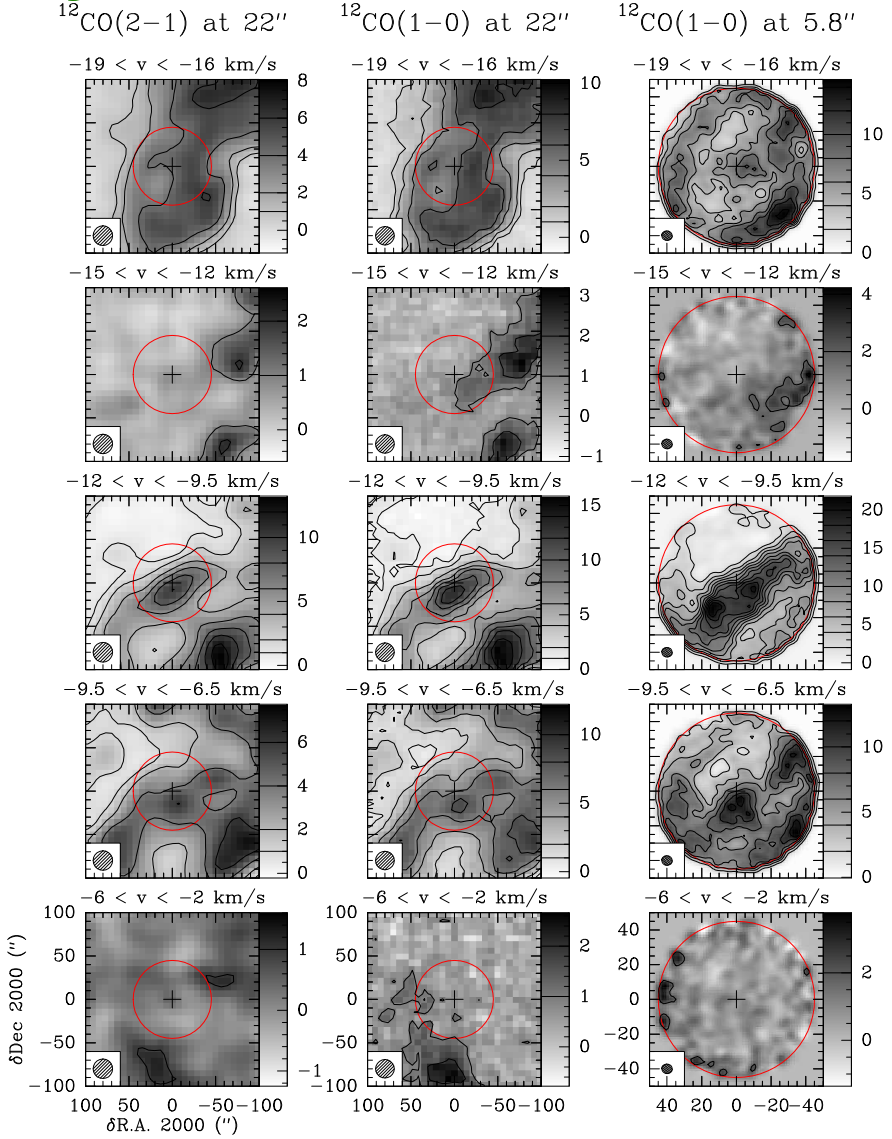
```
MAPPING> let name 1mm
MAPPING> let type lmv-clean
MAPPING> go bit
MAPPING> go flux
Total Flux 461 mJy
```



Science: I. Plateau de Bure interferometer observations of the disk and outflow of HH 30 (Pety et al. A&A 2006, 458, 841)



Science: II. Imaging galactic diffuse gas: bright, turbulent CO surrounding the line of sight to NRAO150 (Pety et al. A&A 2008, 489, 217)



Bug report: I. Wrong way

Hi,

I have just stumbled on an obnoxious bug which prevents me from making the discovery of the century. I will defend my PhD thesis tomorrow. Fix this bug in the coming minutes.

Toto.

Bug report: I. Right way

Dear Gildas team,

Your software is great. For the first time in my life, I encountered a segmentation fault using it. I succeeded to reproduce the bug with a simple list of commands. I attach the following information: version of gildas I am currently using, list of commands and the data set to reproduce the bug. I hope this will help you solve the bug in the coming months. Continue the great work.

Best regards, Toto.

****

gildas version: `dev (07oct08 13:45) (x86_64-fedora6-ifort) source tree`

List of commands:
`MAPPING>` let name 1mm
`MAPPING>` go image
Blablablabla...
Segmentation fault

Data set attached: 1mm.uvt

****