

CASSIS as a plug-in tool in HIPE 10.0

Charlotte Vastel
David Teyssier



To install, select Plug-ins from the Tools menu. Click Install new and enter the following URL:

http://cassis.irap.omp.eu/download/cassis_3.2.jar

If you have an old version of this plugin, please remove it (using "File" "Remove completely" on the plug-ins panel) and install the new version.

Open Hipe 10 with enough memory:

```
java.vm.memory.max=5120m
```

```
hcss.ia.pal.pool.lstore.dir=/Users/vastel/.hcss/lstore
```

Open the oribar_ALL_clean.fits file through the navigator or through the Console:

```
oribar_ALL_clean = fitsReader(file = '/Users/vastel/DATA/HIFI/ORION_BAR/oribar_ALL_clean.fits')
```

It creates a variable: oribar_ALL_clean. Right click on the oribar_ALL_clean variable and "open with" "CASSIS Spectrum1dLineAnalysis"

select HIFI

Click on CO

The screenshot shows the HIPE 10.0.0 - oribarV interface. Annotations include:

- A blue circle around the 'Telescope' dropdown menu, which is set to 'hifi'.
- A blue circle around the 'CO, v=0' entry in the 'Template' list.
- A blue circle around the 'max' field in the 'Threshold' section, which is set to '2000.0 K'.

The 'Template' list shows the following entries:

Name	Tag
H2CN	28503
CO, v=0	28503
HCNH+	28504
C-13-N-15	28505
HCN-15, v=0	28506
HCN-15, v2=1	28507
DNC	28508
DCN, v=0	28509
DCN, v2=1	28510
HC-13-N, v2=1	28511
CH2NH	29003

The 'Parameters' section shows the following values:

- Telescope: hifi
- Mode: Full TE
- Interacting: ☒
- Geometry: Sphere
- Tbg [K]: 2.73
- N(H₂) [cm⁻²]: 7.5E22
- V_{lsr}: 0.0
- Continuum: Continuum 0 [K]

The 'Console' window shows the following commands:

```
HIPE> oribarV = fitsReader(file = '/Users/vastel/DATA/HIFI/ORION_BAR/oribarV.fits')
HIPE> # Added variable: lineAnalysis_0
HIPE> # Added variable: cassisLineProduct_1
HIPE> del(cassisLineProduct_1)
HIPE> # Added variable: cassisLineProduct_2
HIPE> del(cassisLineProduct_2)
HIPE> # Added variable: cassisLineProduct_3
HIPE>
```

Select the upper energy: 2000K means that it will look for all the transitions for energy lower than 2000K

press DISPLAY

HIPE 10.0.0 - oribarV

File Edit Run Pipelines Window Tools Help

data_reduction.py x fitBaseline x hiClass x hifiPipeline x flagChanger.py x fitBaseline x doDeconvolution x flagTool x oribarV x lineAnalysis_0 x cassisLineProduct_3 x

File Edit Models View Scripts Windows Templates Configuration VO Help

Data

Load UNKNOWN Vlsr data: 0.0 km/s in: SKY Telescope hifi

Tuning

Range min: 479.514 max: 1906.773 GHz Band: 60.0 km/s

Threshold

Eup min: 0.0 max: 2000.0 K Aij min: 0.0 max: *
Jup min: * max: * Kup min: * max: * Lup min: * max: * Mup min: * max: *

LTE-RADEX ☐

Parameters

Telescope: hifi ☐ Tmb->Ta conv hifi

Noise rms: 0.0 mK

Oversampling Oversampling: 3.0

Template

Name	Tag
H2CN	28502
CO, v=0	28503
HCNH+	28504
C-13-N-15	28505
HCN-15, v=0	28506
HCN-15, v2=1	28507
DNC	28508
DCN, v=0	28509
DCN, v2=1	28510
HC-13-N, v2=1	28511
CH2NH	29003

Load config

Display

Save config

Component 1 ☒ +

Mode: Full LTE

Molecules: -- Operations --

☒ Interacting

Geometry: Sphere

Tbg [K]: 2.73

N(H₂) [cm⁻²]: 7.5E22

V_{lsr}: 0.0 km/s

Continuum Continuum 0 [K]

Species	Tag	Database	Compute	N(Sp) (cm ⁻²)	Abundance (/H2)	Tex (K)	FWHM (km/s)	Size (")
CO, v=0	28503	CDMS	<input checked="" type="checkbox"/>	7.00E14	1.00E-8	100.00	1.00	3.00

History Log Console x

```
HIPE> oribarV = fitsReader(file = '/Users/vastel/DATA/HIFI/ORION_BAR/oribarV.fits')
HIPE> # Added variable: lineAnalysis_0
HIPE> # Added variable: cassisLineProduct_1
HIPE> del(cassisLineProduct_1)
HIPE> # Added variable: cassisLineProduct_2
HIPE> del(cassisLineProduct_2)
HIPE> # Added variable: cassisLineProduct_3
HIPE>
```

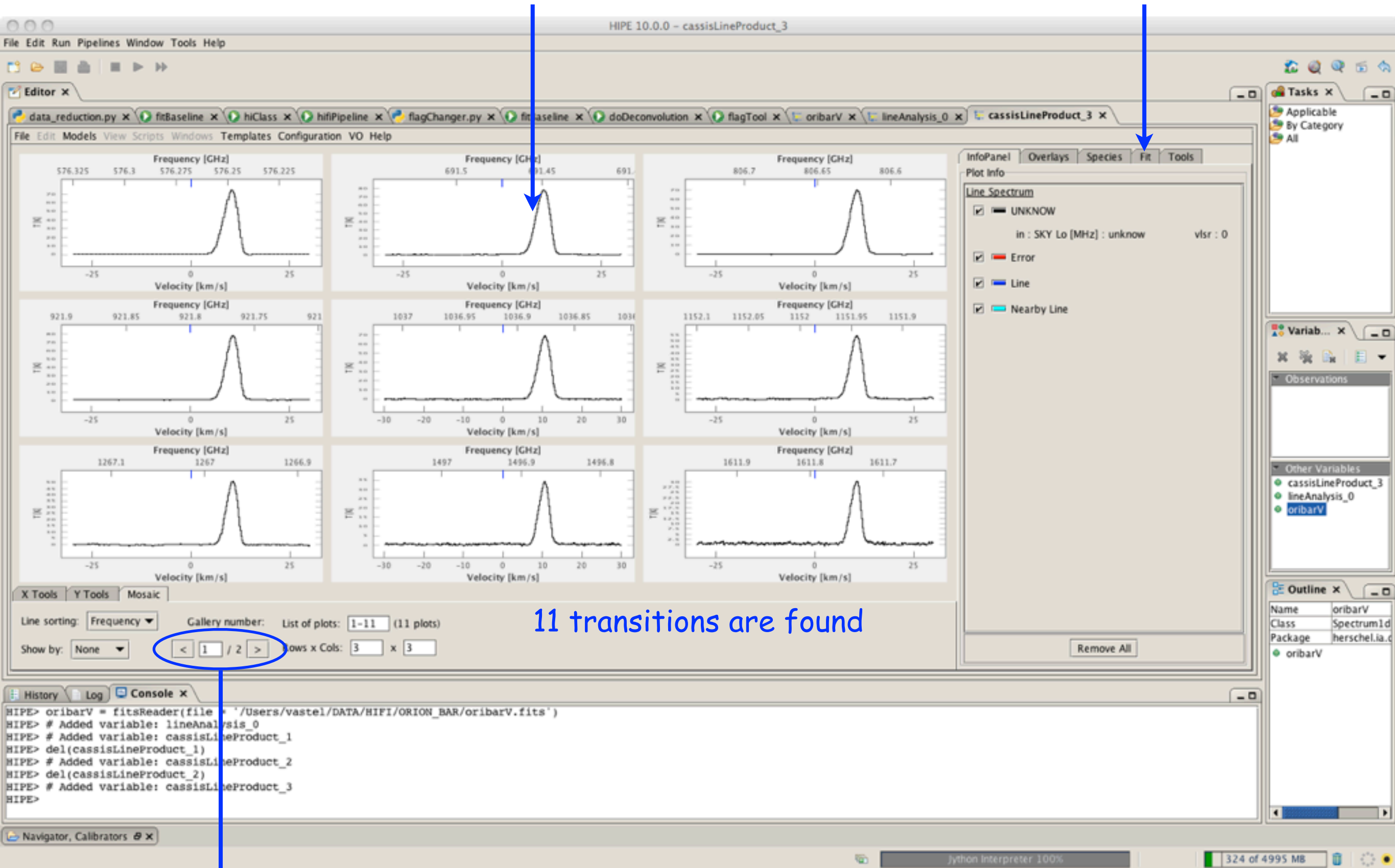
Navigator, Calibrators x

Python Interpreter 100%

324 of 4995 MB

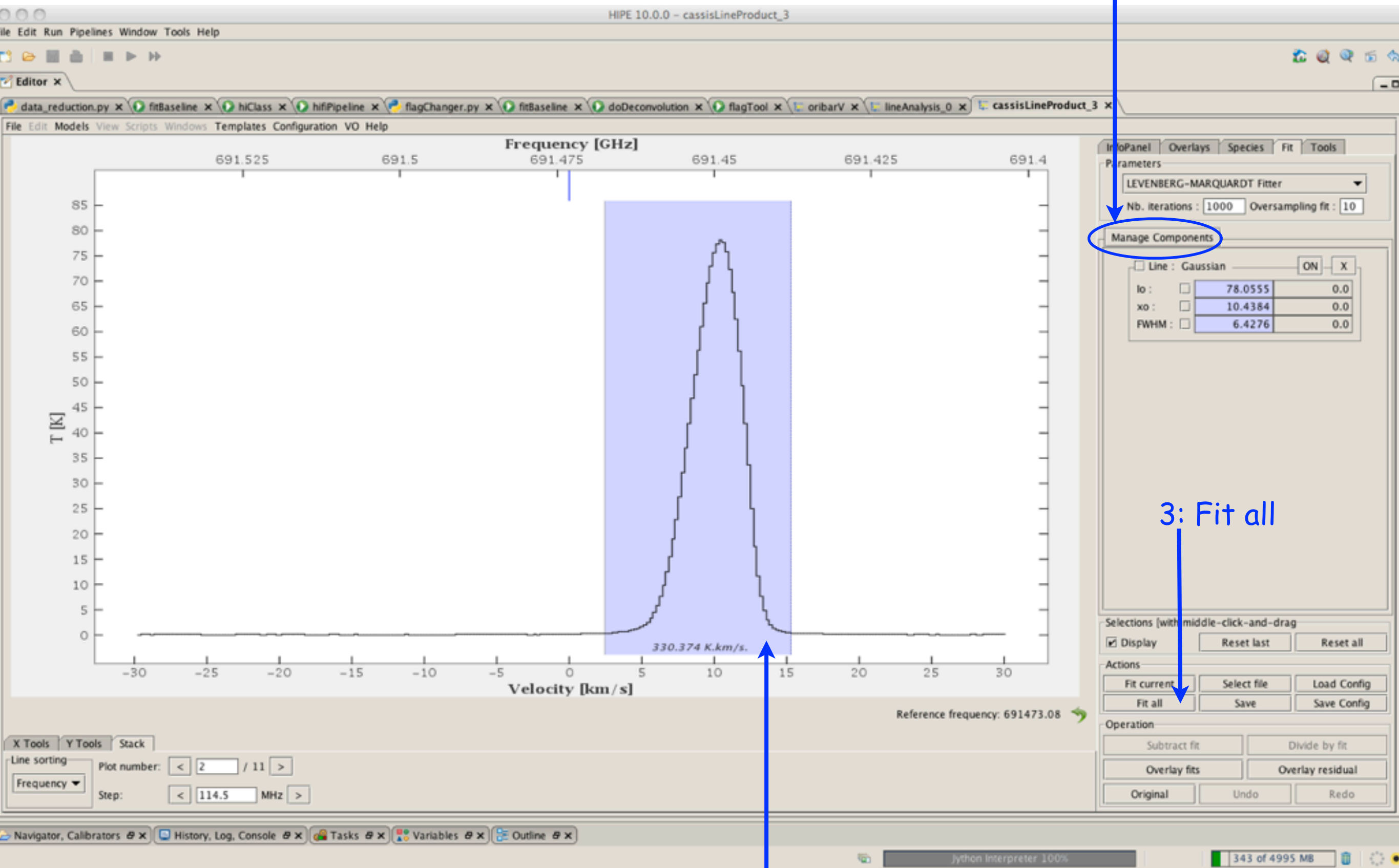
Double-Click on 1 of the transition

Click on Fit



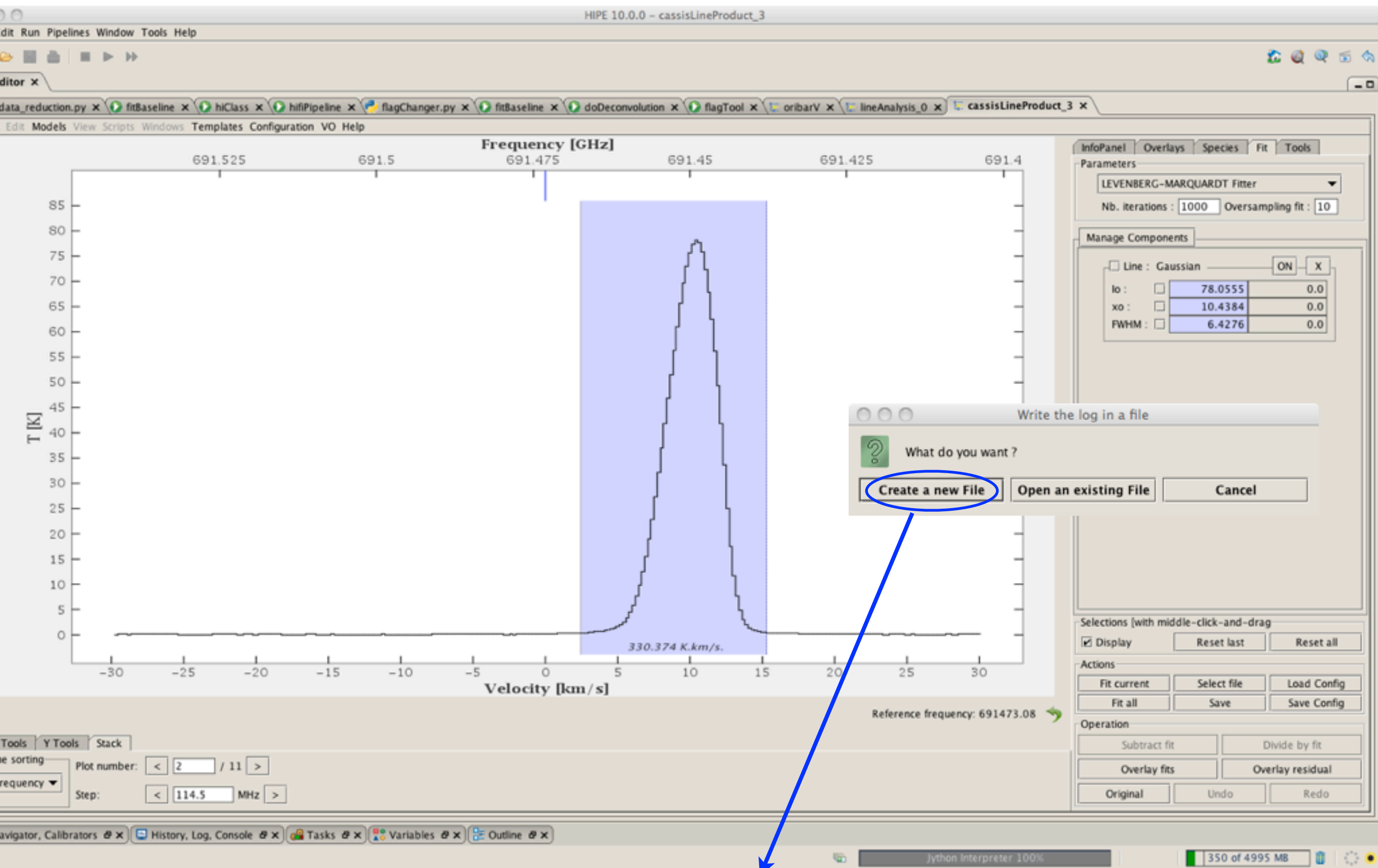
browse the transition from 1 window to another

1: Select Gaussian (the baselines are already subtracted in these data, but you can also fit the continuum)

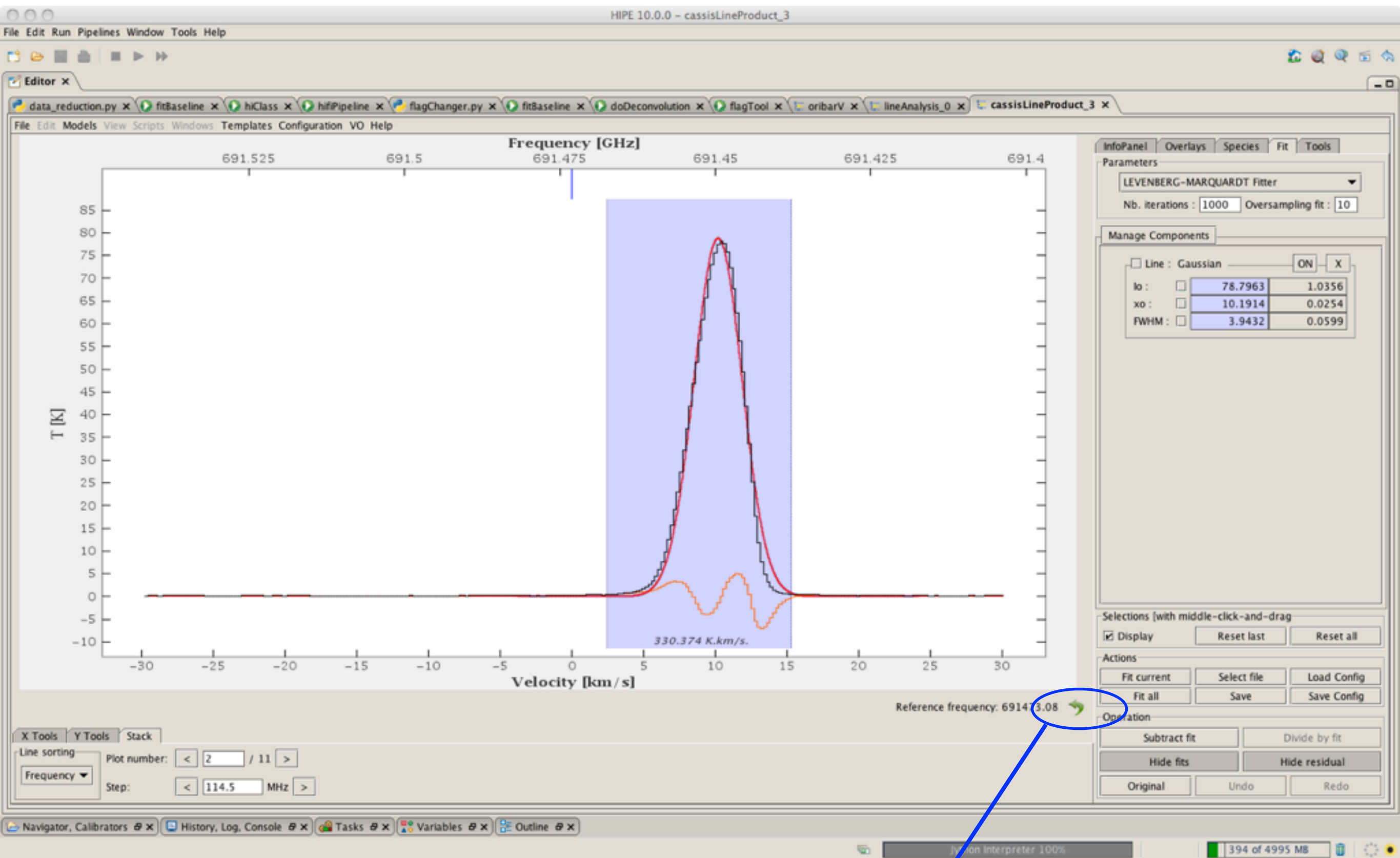


3: Fit all

2: Use your mouse (middle-click, or alt-left click if you don't have a mouse) to select the area to be fitted (here in blue)

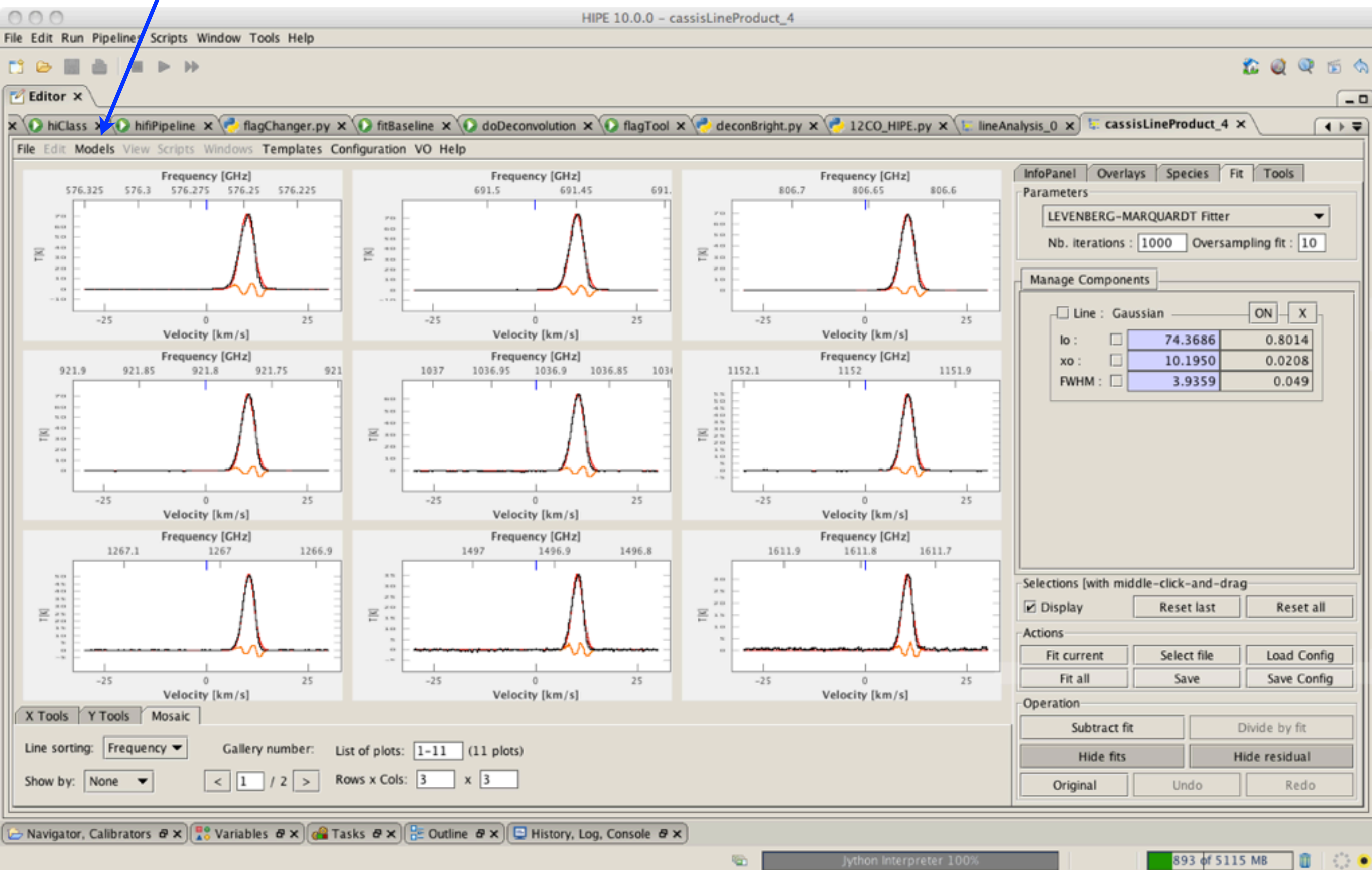


Click on "Create a new file": it will save all the fits (in this case, 1 Gaussian fit per transition, i.e. 11 fits) in the same file



Click on the green arrow to see all the fits

After admiring your beautiful fits, click on "Models", then "Rotational Diagram"



open the file that you created in Slide 7 (where the fits where saved)

The screenshot shows the HIPE 10.0.0 - rotationalView interface. The main plot area displays a graph with the y-axis labeled $\ln(Nu/gu)$ ranging from 0.00 to 1.05 and the x-axis labeled $Eup/k[K]$ ranging from 0.00 to 1.05. The plot area is currently empty, with the text "No data" centered. The right-hand side of the interface contains a control panel with several sections:

- Select/Write:** Contains buttons for "Select datafile" (circled in blue), "Write datafile", and status indicators "No file selected" and "No Data written".
- Build diagram from:** Contains a dropdown menu currently set to "1st moment".
- Scaling parameters:** Contains checkboxes for "Ta->Tmb conv" and "Beam dilution correction", both of which are circled in blue. Below them is a "Source Size [']:" input field with the value "0".
- Display:** Contains a checked checkbox for "Display links" and a "Display" button (indicated by a blue arrow).

At the bottom of the interface, there is a toolbar with various tools and a status bar at the very bottom showing "jython Interpreter 100%" and "409 of 4995 MB".

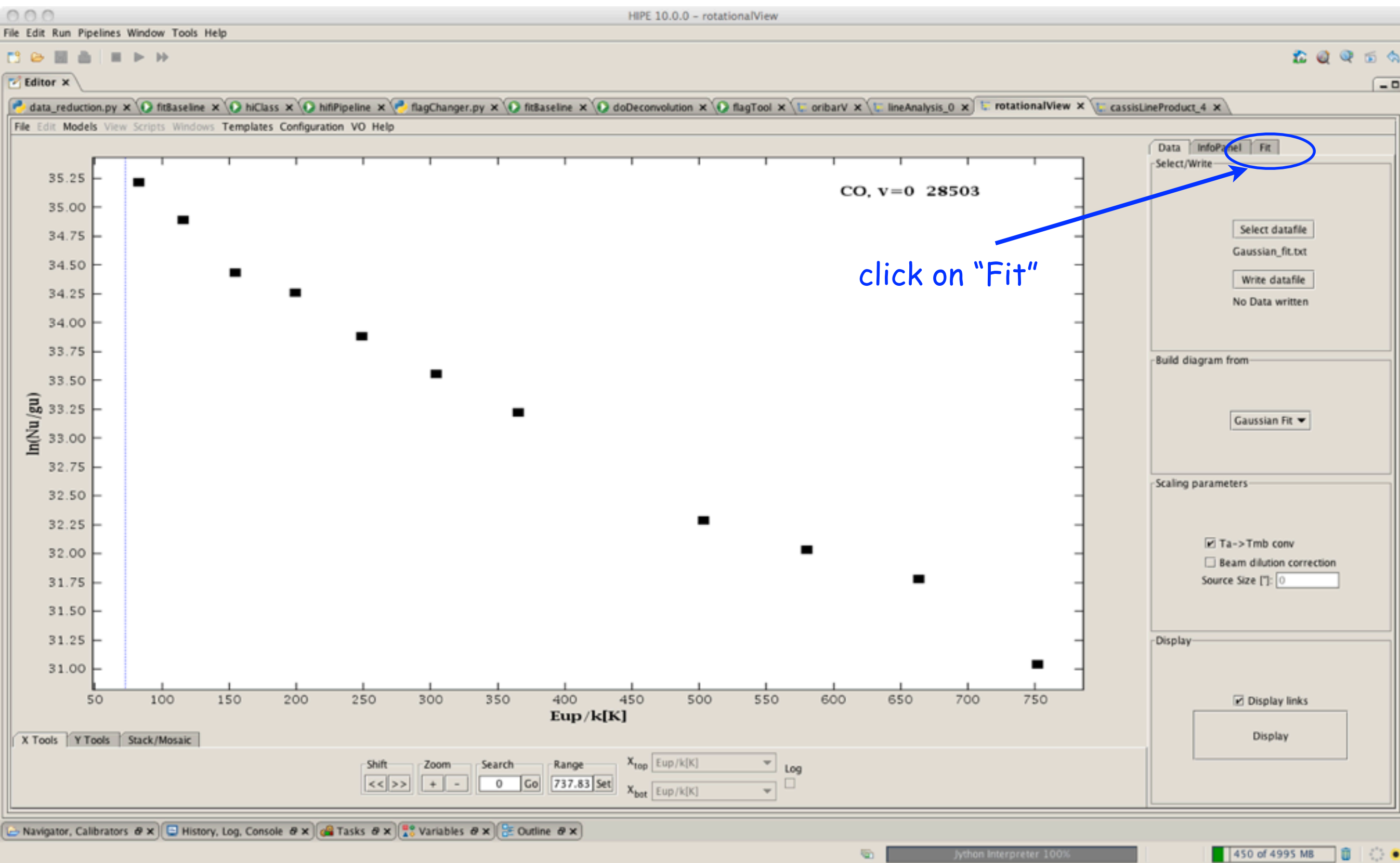
To compute the rotational diagram, you can either use the Gaussian Fit, or the integrated area from the (blue) selection ("1st moment") (see slide 6)

You can:

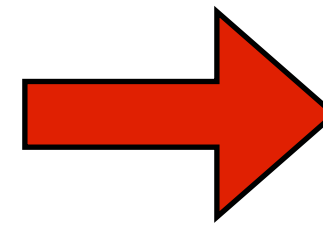
- Apply a "Ta to Tmb" conversion
- Apply a beam dilution correction

Click on Display after the above selection

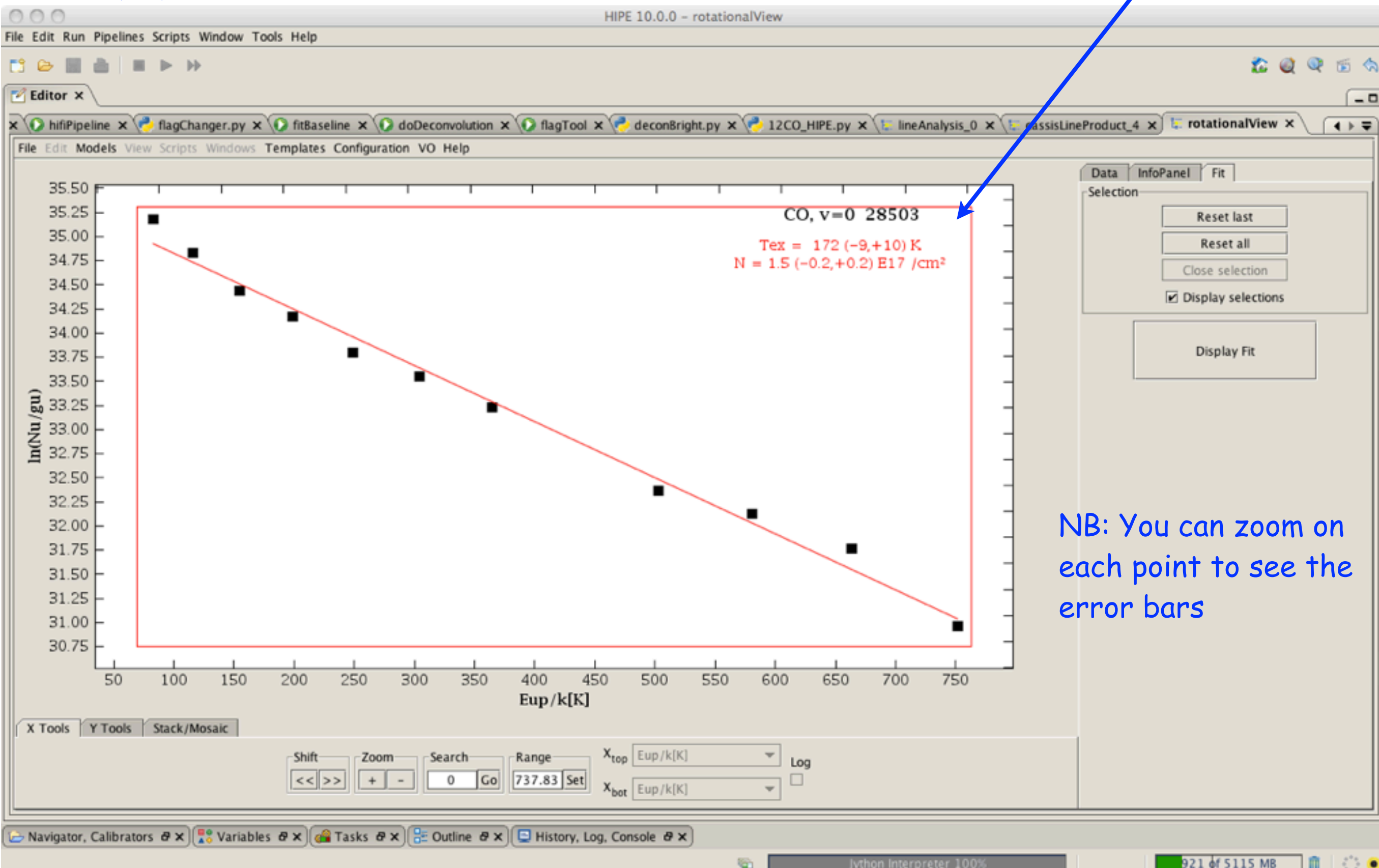
Here are the 11 transitions. You can double-click on each transition to see the informations



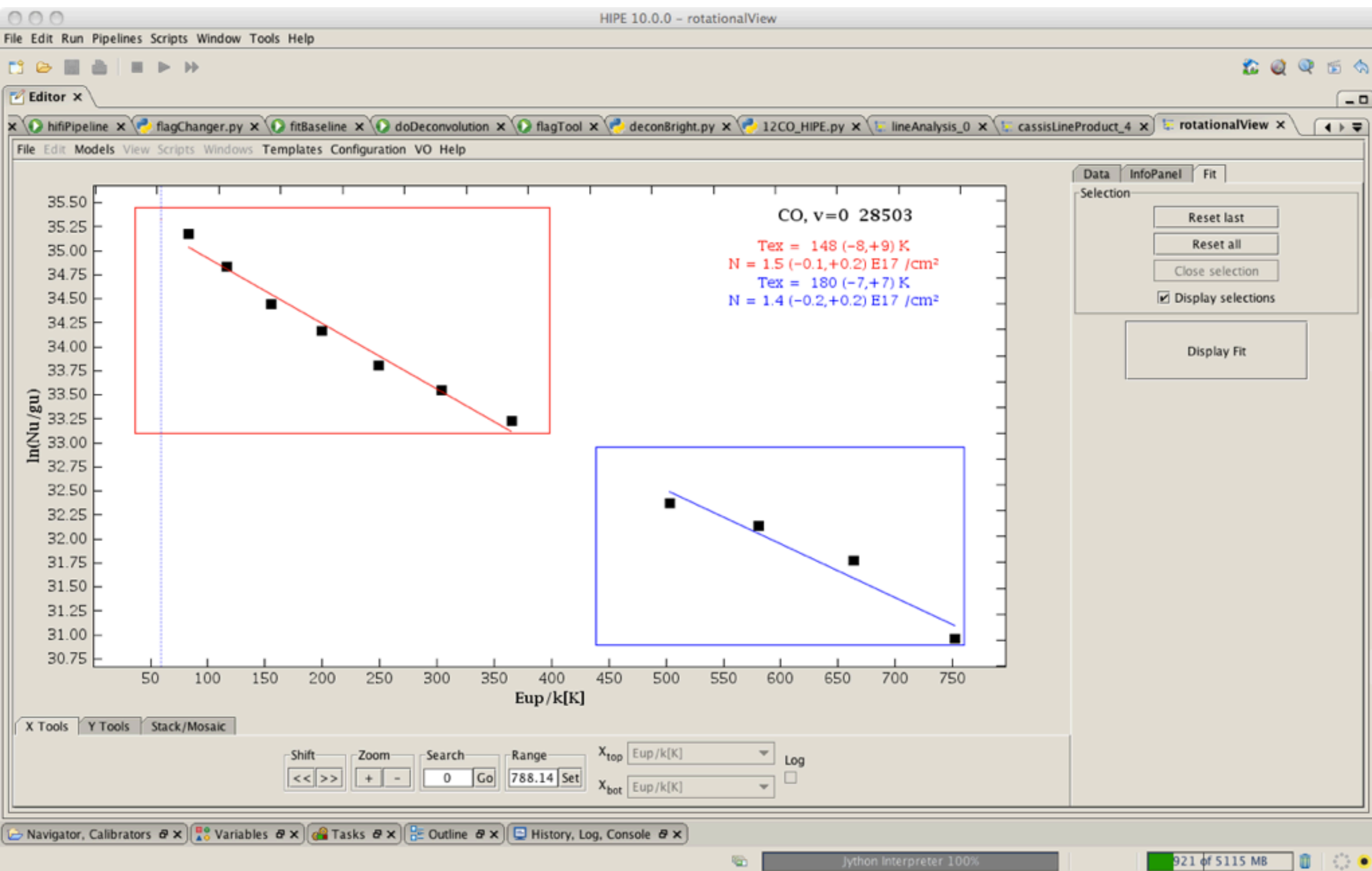
- 1: Select all your points (or some of them, by using your middle-click),
- 2: Click on "Close selection"
- 3: "Display Fit"



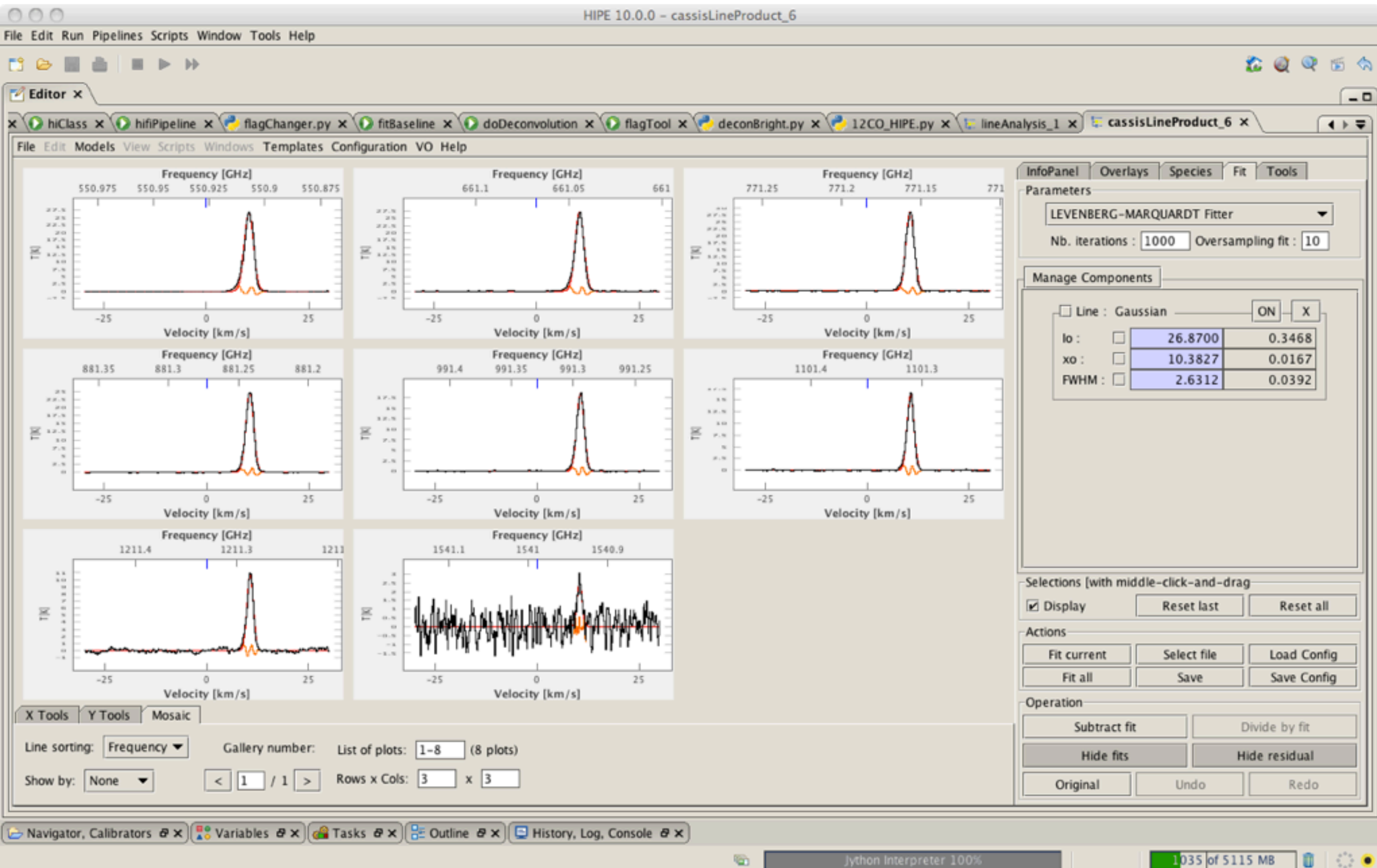
results from the rotational diagram



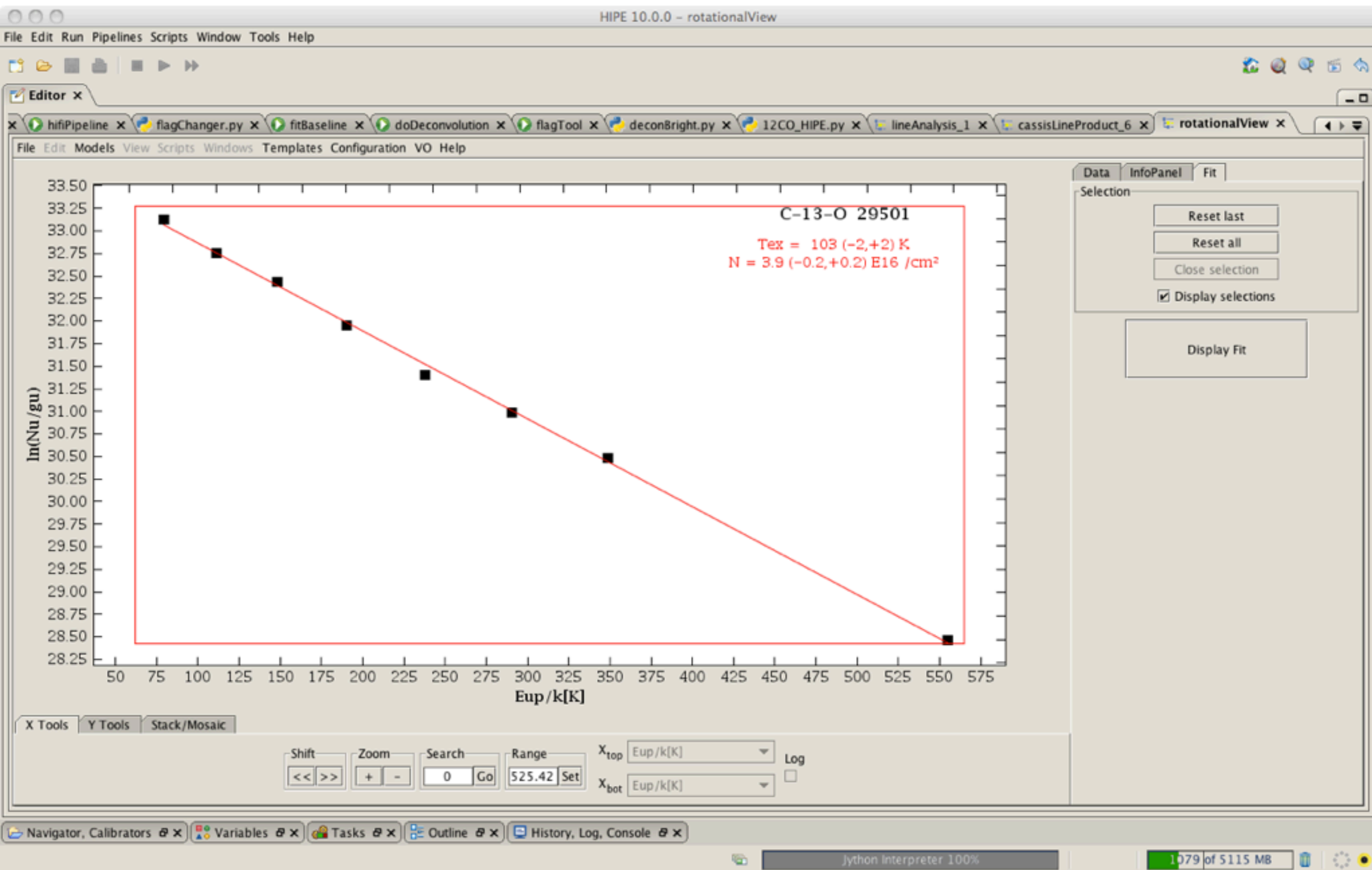
You can also decide for different components. Below 2 components with different T_{ex} and N .



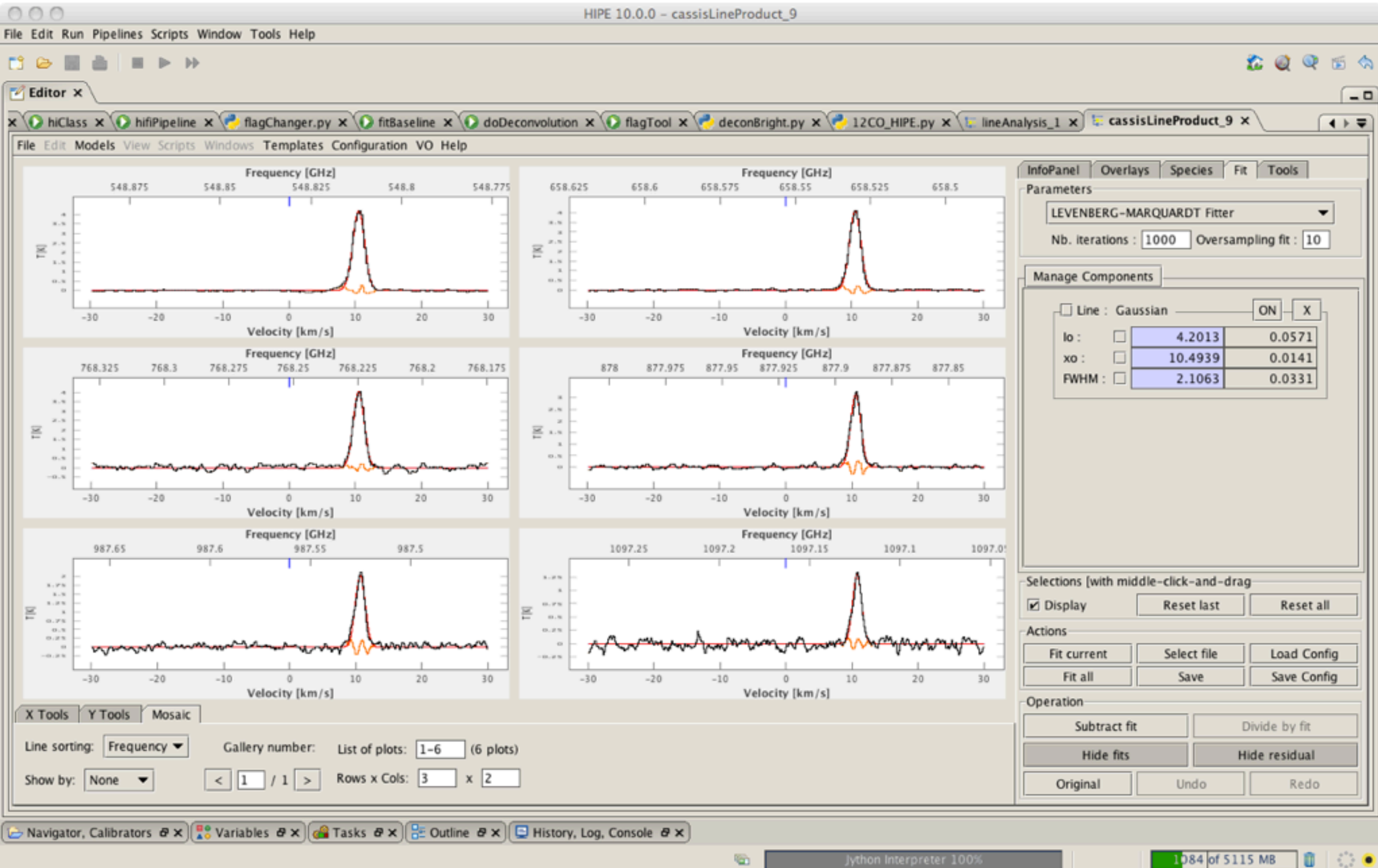
Same thing for ^{13}CO (select Eup lower than 630 K)



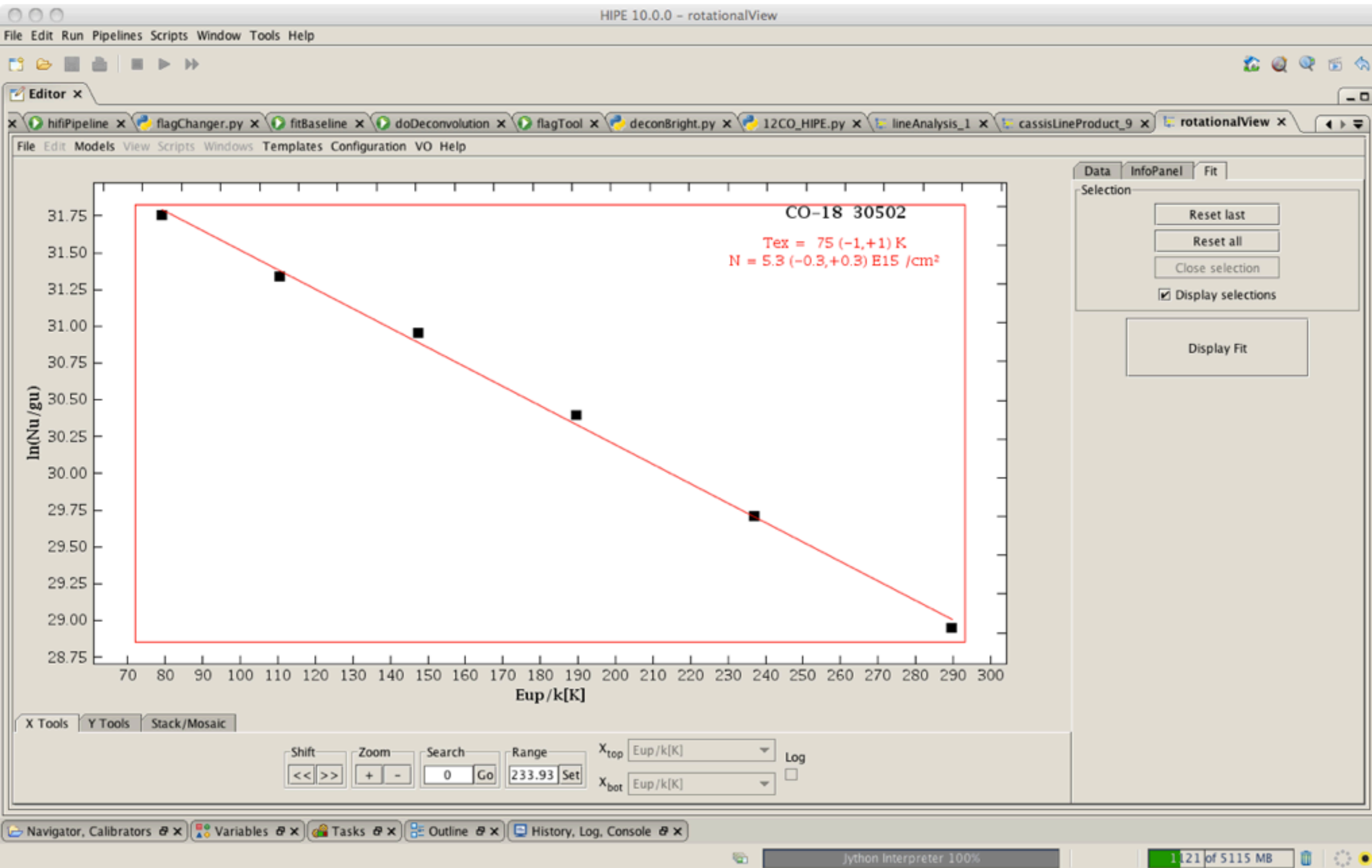
results for ^{13}CO



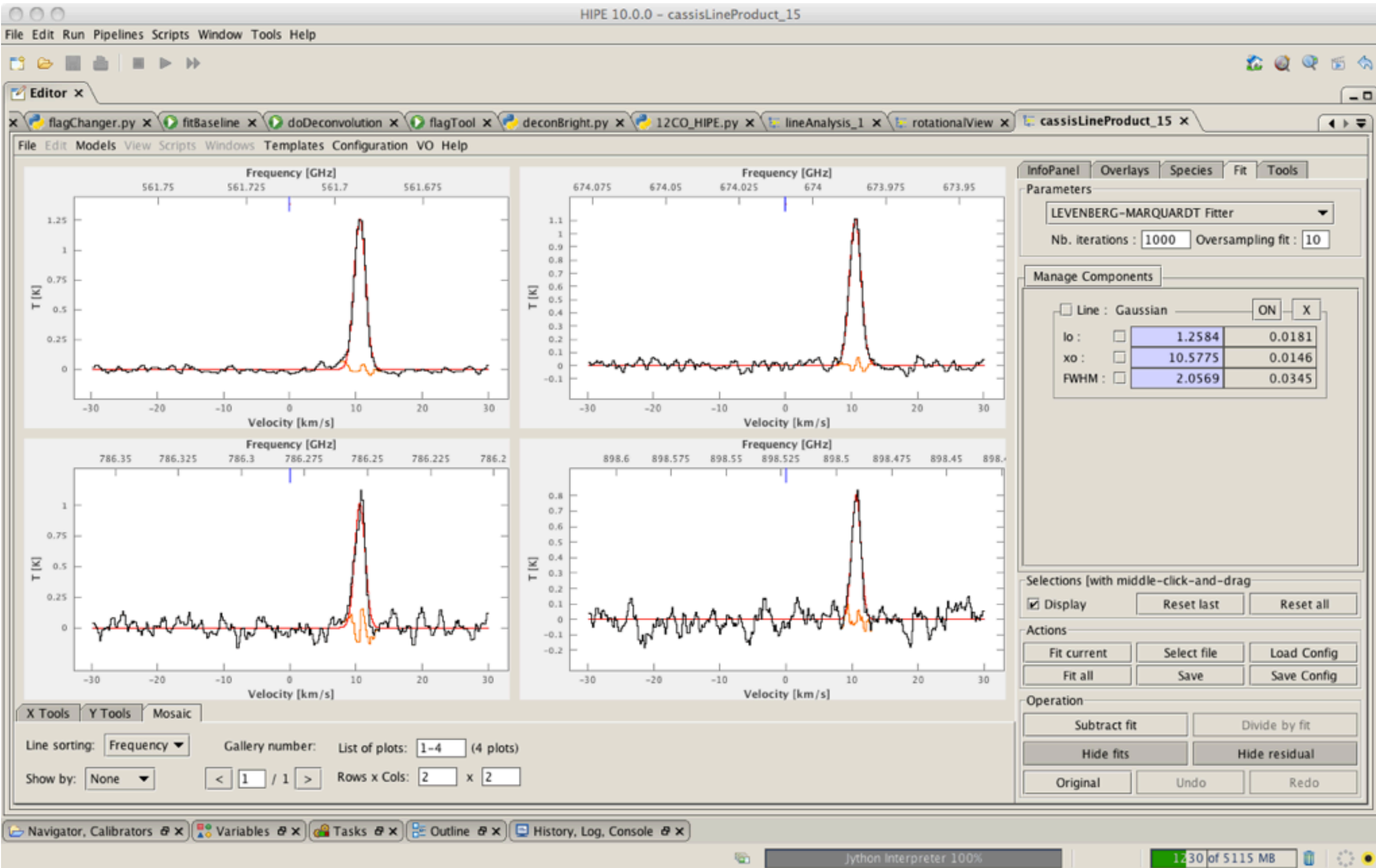
Same thing for $C^{18}O$ (select Eup lower than 300 K)



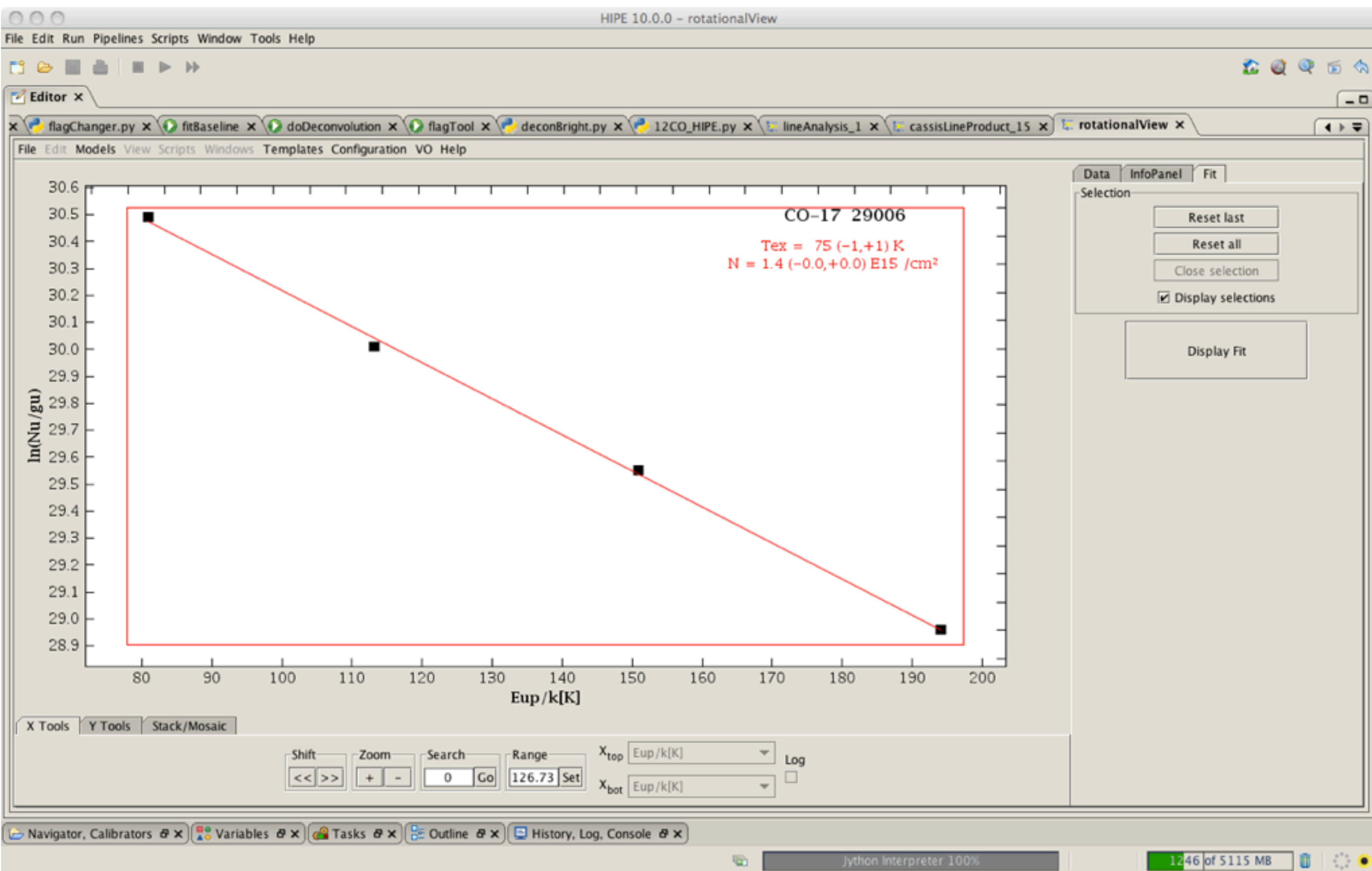
results for $C^{18}O$



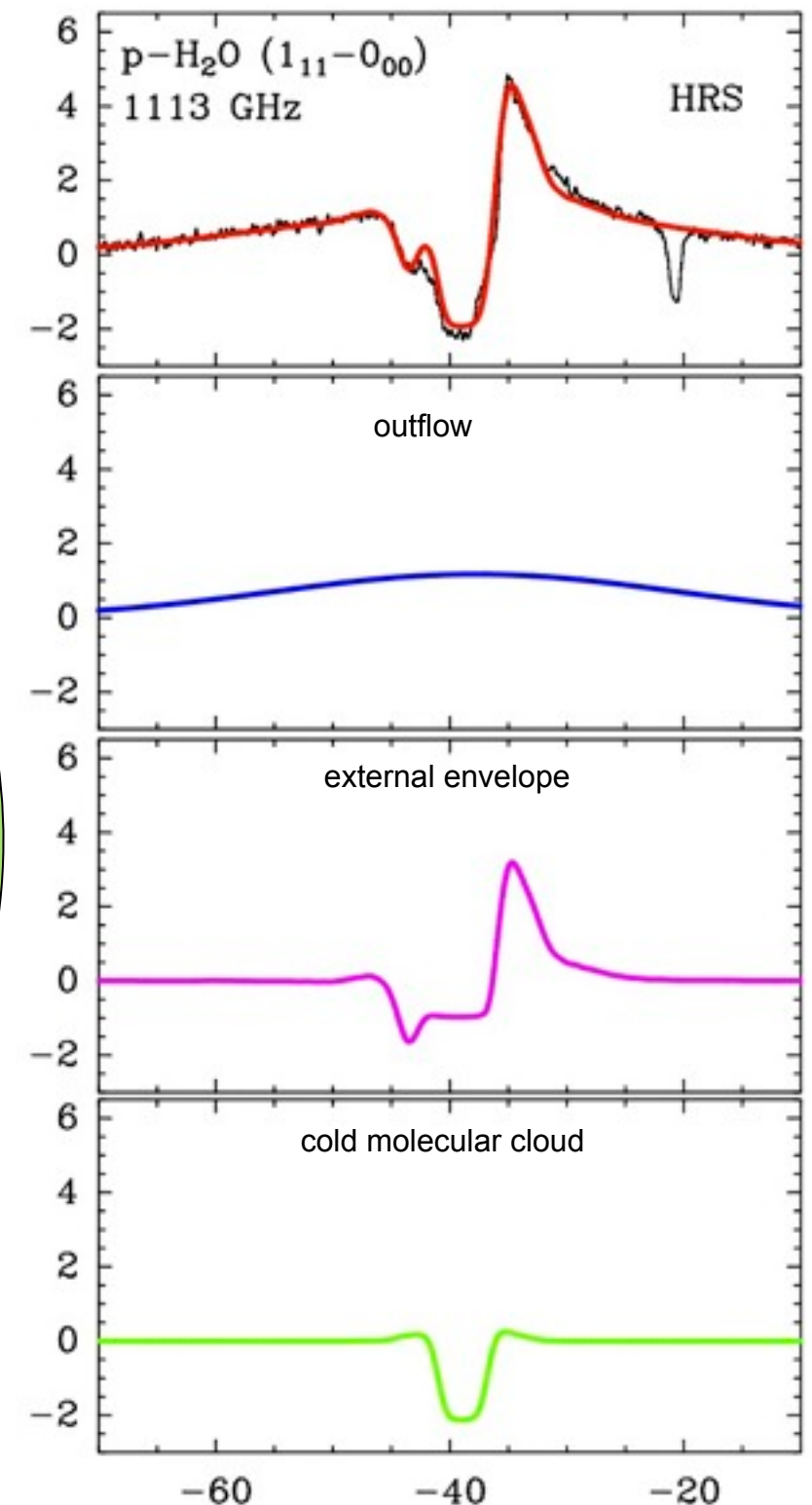
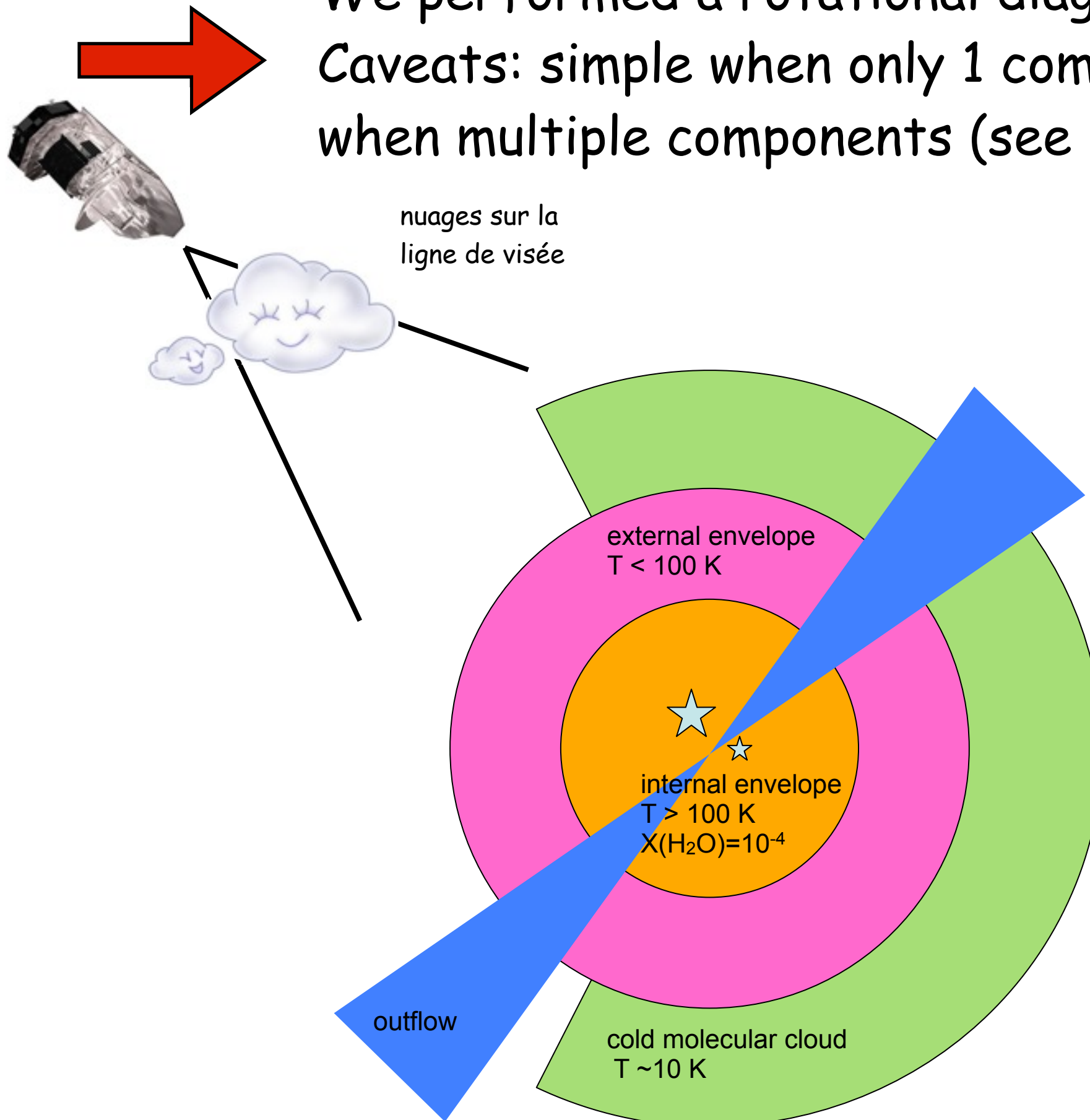
Same thing for $C^{17}O$ (select Eup lower than 300 K), with JPL database (no hyperfine structure)

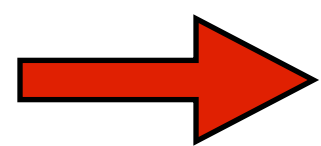


results for $C^{17}O$

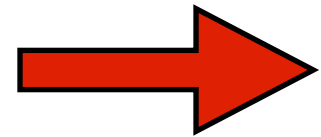


We performed a rotational diagram analysis.
Caveats: simple when only 1 component, but impossible when multiple components (see below)





Then we might need to disentangle the many components, before doing the rotational diagram fitting for each of the components.



Also, we might need to explore the many parameters needed for complete radiative transfer modeling: excitation temperature, column density, velocity in the local standard of rest, linewidth, beam dilution, interaction with the continuum, (and H₂ density, kinetic temperature for non-LTE modeling). Then you must explore all those parameters (beware that it can be time consuming) within grids of parameters spacing. That can be done within CASSIS with the χ^2 minimization. The best fit can be found when the χ^2_{red} is minimum.

for each spectrum

$$\chi_i^2 = \sum_{j=1}^N \frac{(I_{obs,j} - I_{model,j})^2}{rms_i^2 + cal_i^2 (I_{obs,j} - I_{cont,j})^2}$$

N: number of channels in the spectrum

rms: noise in the spectrum

cal: calibration factor

I_{obs}: observed intensity within a channel

I_{model}: modeled intensity within a channel

taking into account
all spectra

$$\chi_{reduit}^2 = \frac{1}{N_{spectre} \sum_{i=1}^{N_{spectre}} w_i} \sum_{i=1}^{N_{spectre}} \frac{\chi_i^2 w_i}{\frac{N}{N_{ind}} - dof}$$

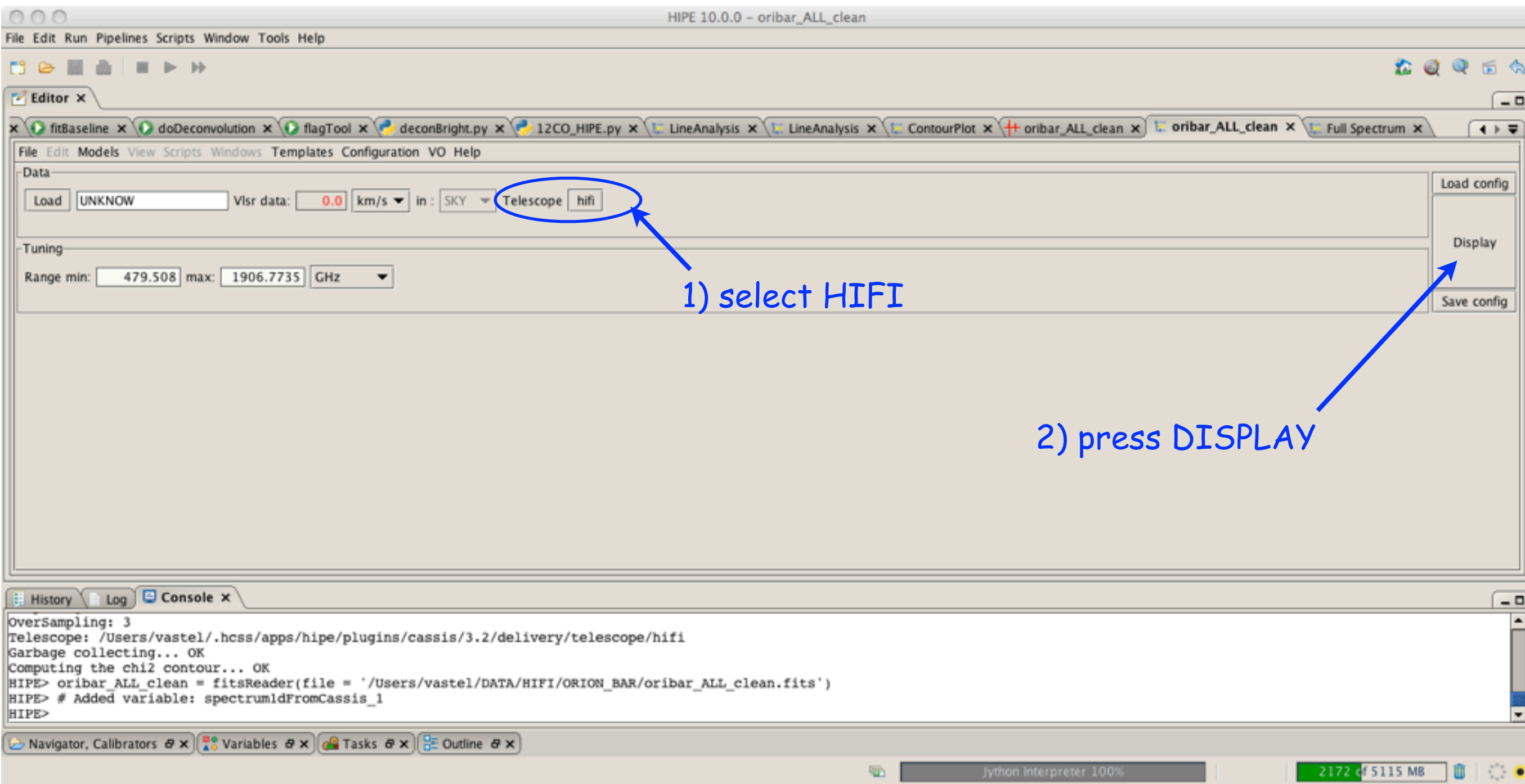
N_{spectre}: total number of spectra

N_{ind}: total number of independent points

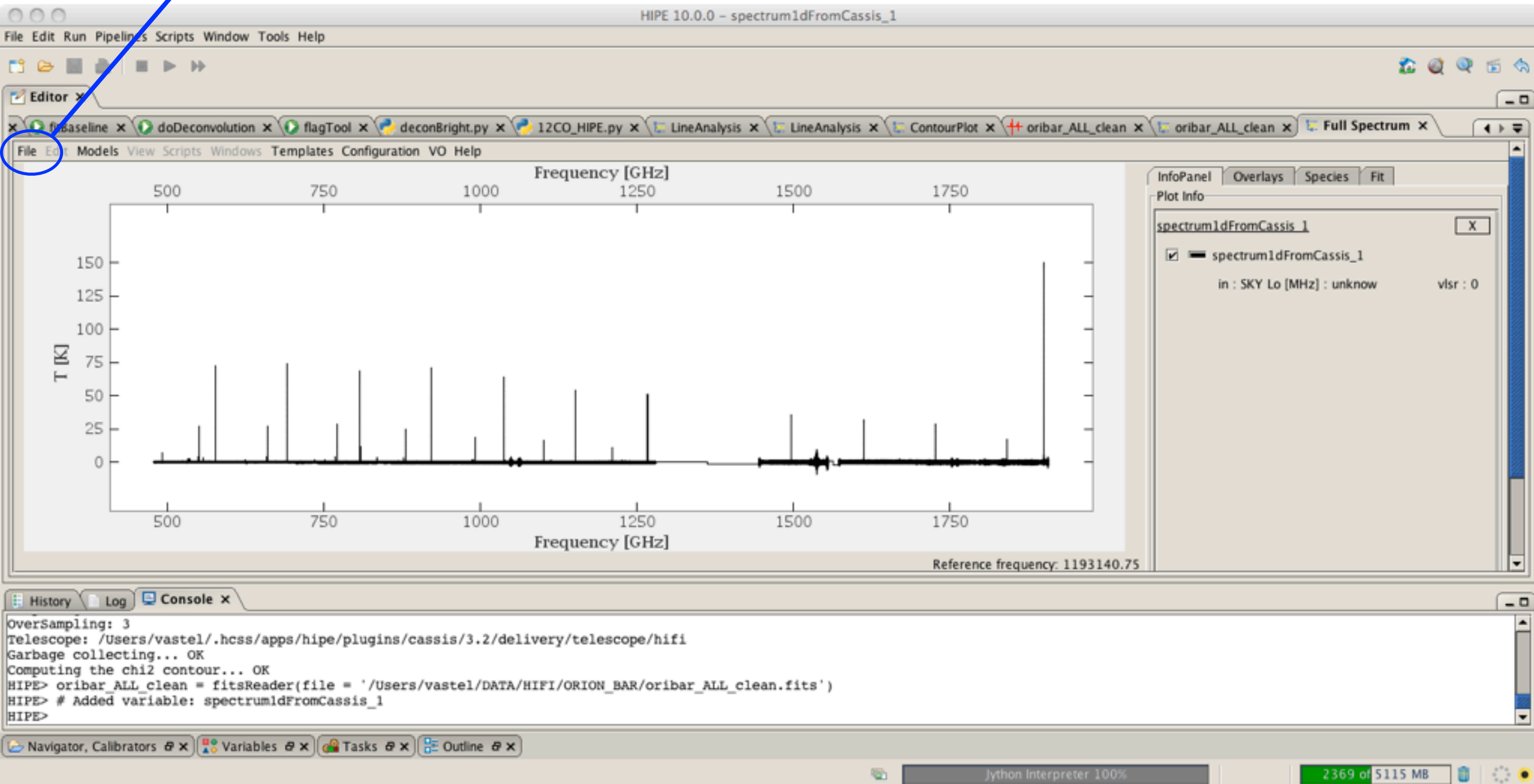
dof: degree of liberty

w_i: weight for each spectrum

Right click on the oribar_ALL_clean variable and "open with"
"CASSIS Spectrum1dAnalysis"



"File", "Save"



Save your file as a .fus file

The screenshot displays the HIPE software interface. The main window shows a spectral plot with 'T [K]' on the y-axis (0 to 150) and 'Frequency [GHz]' on the x-axis (500 to 1750). A blue arrow points from the text 'Save your file as a .fus file' to the 'Files of Type' dropdown in the 'Save' dialog, which is set to '*.fus'. The 'Save' dialog is open, showing the 'ORION_BAR' folder and the file 'spectrum1dFromCassis_13.fus'. The 'File Name' field contains 'spectrum1dFromCassis_oribar_ALL_clean'. The 'InfoPanel' on the right shows 'spectrum1dFromCassis_1' with a checked box and 'in : SKY Lo [MHz] : unknow' and 'vlr : 0'. The 'Console' at the bottom shows the following output:

```
OverSampling: 3
Telescope: /Users/vastel/.hcss/apps/hipe/plugins/cassis/3.2/delivery/telescope/hifi
Garbage collecting... OK
Computing the chi2 contour... OK
HIPE> oribar_ALL_clean = fitsReader(file = '/Users/vastel/DATA/HIFI/ORION_BAR/oribar_ALL_clean.fits')
HIPE> # Added variable: spectrum1dFromCassis_1
HIPE>
```

The status bar at the bottom indicates 'Jython Interpreter 100%' and '2369 of 5115 MB'.

Open the 12CO_HIPE.py script

The screenshot displays the HIPE 10.0.0 software interface. The title bar indicates the file path: `/Users/vastel/DATA/HIFI/ORION_BAR/12CO_HIPE.py`. The menu bar includes File, Edit, Run, Pipelines, Scripts, Window, Tools, and Help. The toolbar contains various icons, with the 'Open' icon (a folder) circled in blue. A blue arrow points from the text 'Open the 12CO_HIPE.py script' to this icon.

The Editor window shows the `12CO_HIPE.py` script with the following code:

```
1 import ScriptEnvironment
2 import time
3 from Component import Component
4 from Range import Range
5 from ScriptLineAnalysisRG import UserInputs
6
7 # INPUTS
8 # =====
9 Range.unit      = "km/s"          # Possible units are GHz and km/s, MHz, cm-1 and micrometer to be implemented
10 r1             = Range(0.0, 20.0)
11 sourceName     = "Orion Bar "
12 speciesName    = "12CO"
13 myDir          = "/Users/vastel/DATA/HIFI/ORION_BAR/"
14 myName         = sourceName+speciesName
15
16 # =====
17 # USER INPUTS
18 # =====
19 userInputs     = UserInputs(
20 fileName       = myDir+"spectrumIdFromCassis_oribar_ALL_clean",
21 suffixName     = [""],
22 extensionName  = ".fus",
23 telescope      = "/Users/vastel/.hcss/apps/hipe/plugins/cassis/3.2/delivery/telescope/hifi",
24 tuningRange    = [479, 1910],
25 tuningBand     = 60,
26 aijMin         = 1.0e-5,
27 eup            = [0.0, 2000.0],
28 kup            = ["*", "*"],
29 template       = "Full CDMS",
30 moltags        = [28503],
31 tmb2ta         = True,
32 # isoUnique    = False,
33 outputFile     = myDir+myName+".dat"
```

The Console window shows the following output:

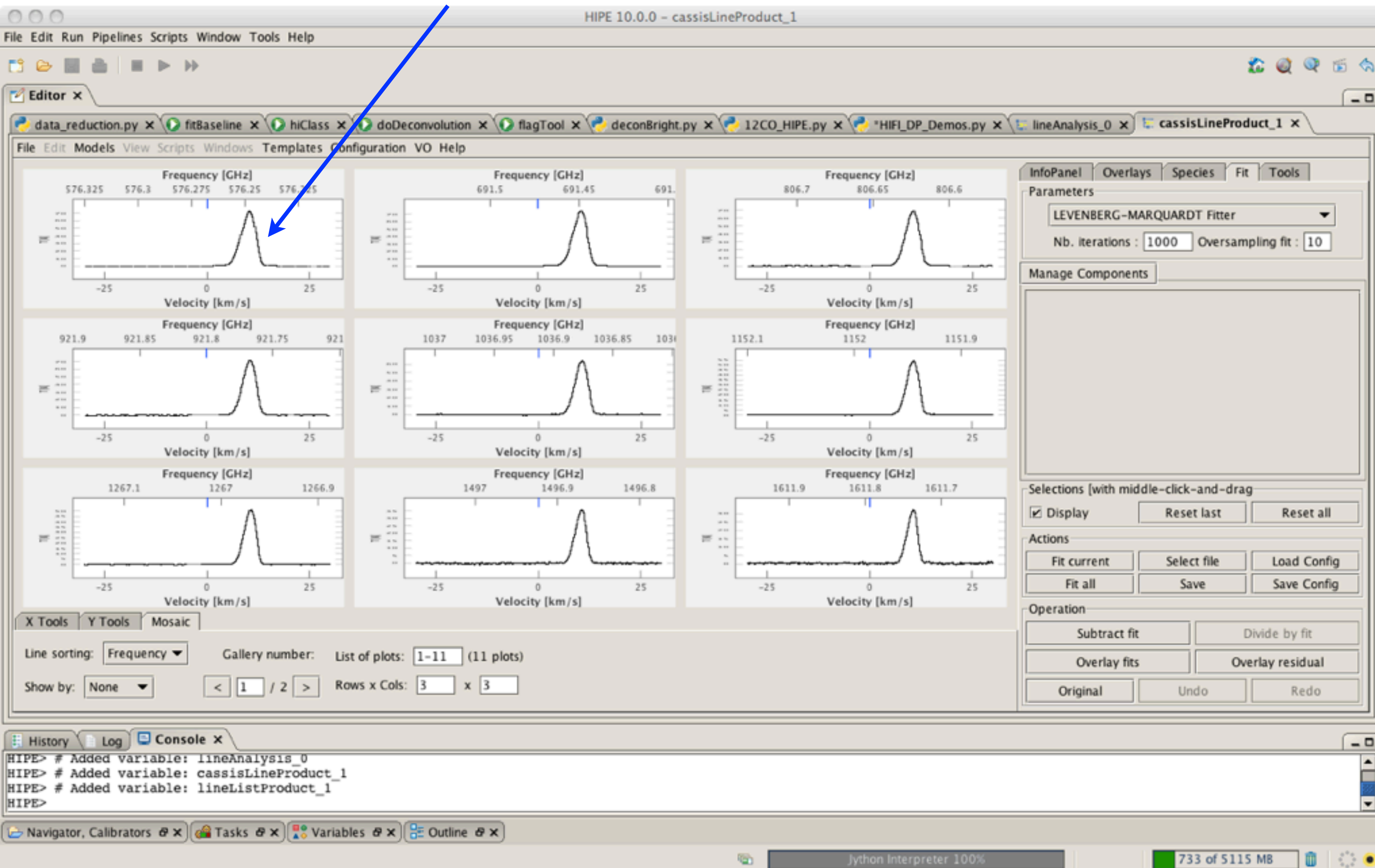
```
Best parameters 1: PhysicalModel(visr = 10.25, tex = 155.556, nmol = 2.22605e+17, size = 300, fwhm = 3.5, tbg = 2.73, model = lte, continuum = continuum-0)
Total nb of points used in the computation = 1753 / 5244
-----
Computing the best model... OK
OverSampling: 3
Telescope: /Users/vastel/.hcss/apps/hipe/plugins/cassis/3.2/delivery/telescope/hifi
Garbage collecting... OK
Computing the chi2 contour... OK
HIPE>
```

The bottom status bar shows the progress of the `12CO_HIPE` task at 100% and the memory usage as 844 of 5115 MB.

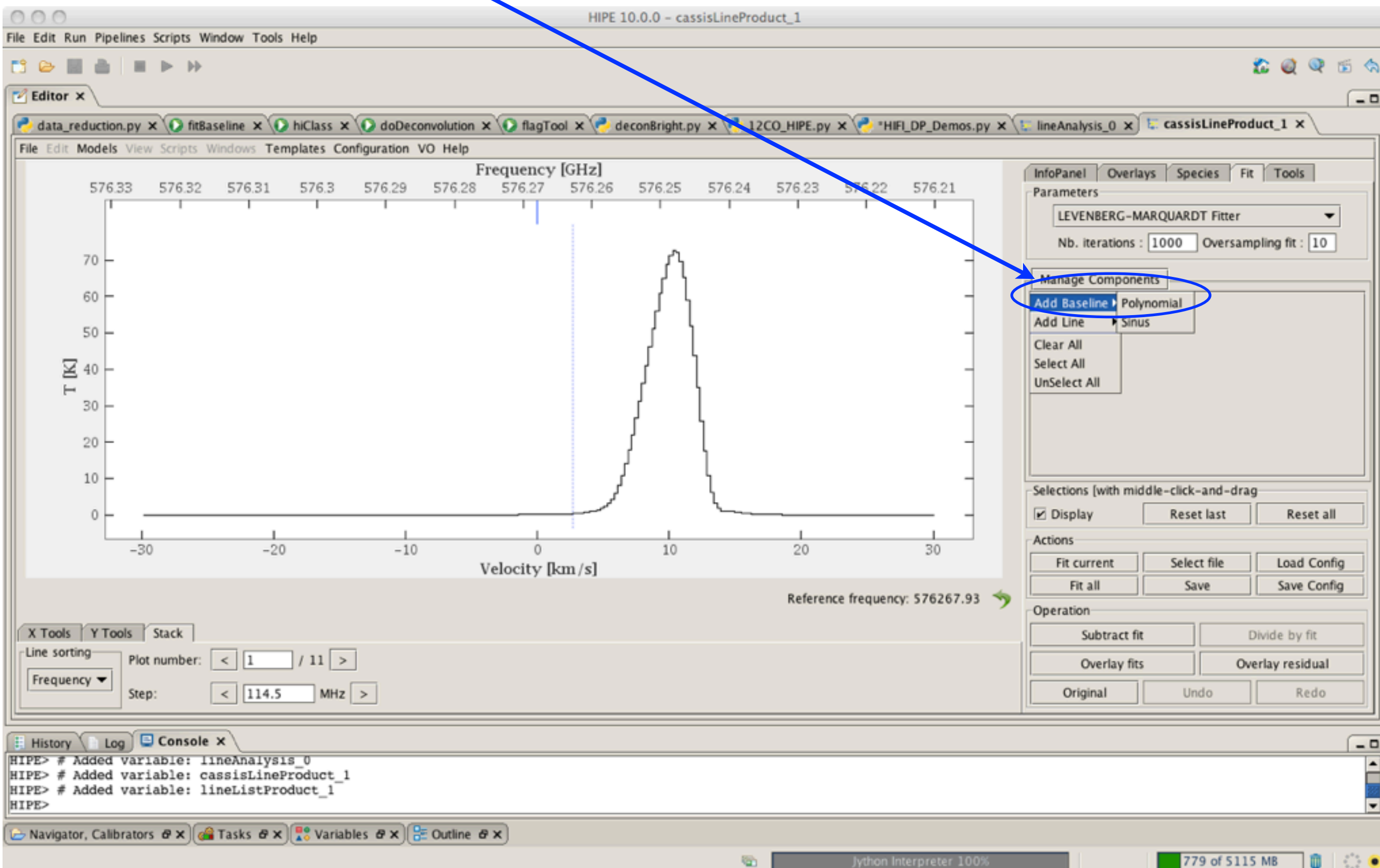
Interactive part with the user to modify the
parameters of the python script

How to compute the rms with CASSIS?

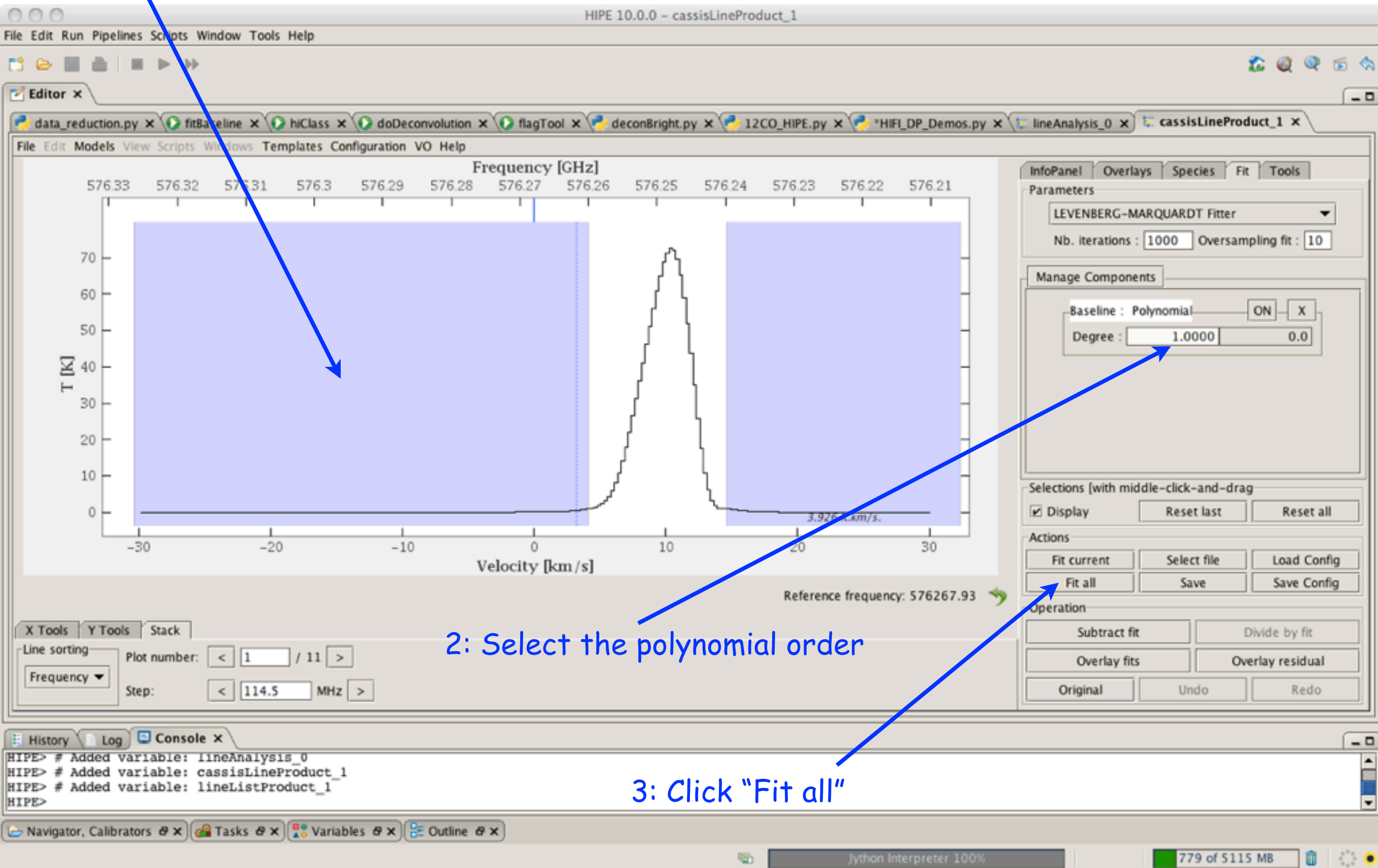
Double-Click on 1 of the transition



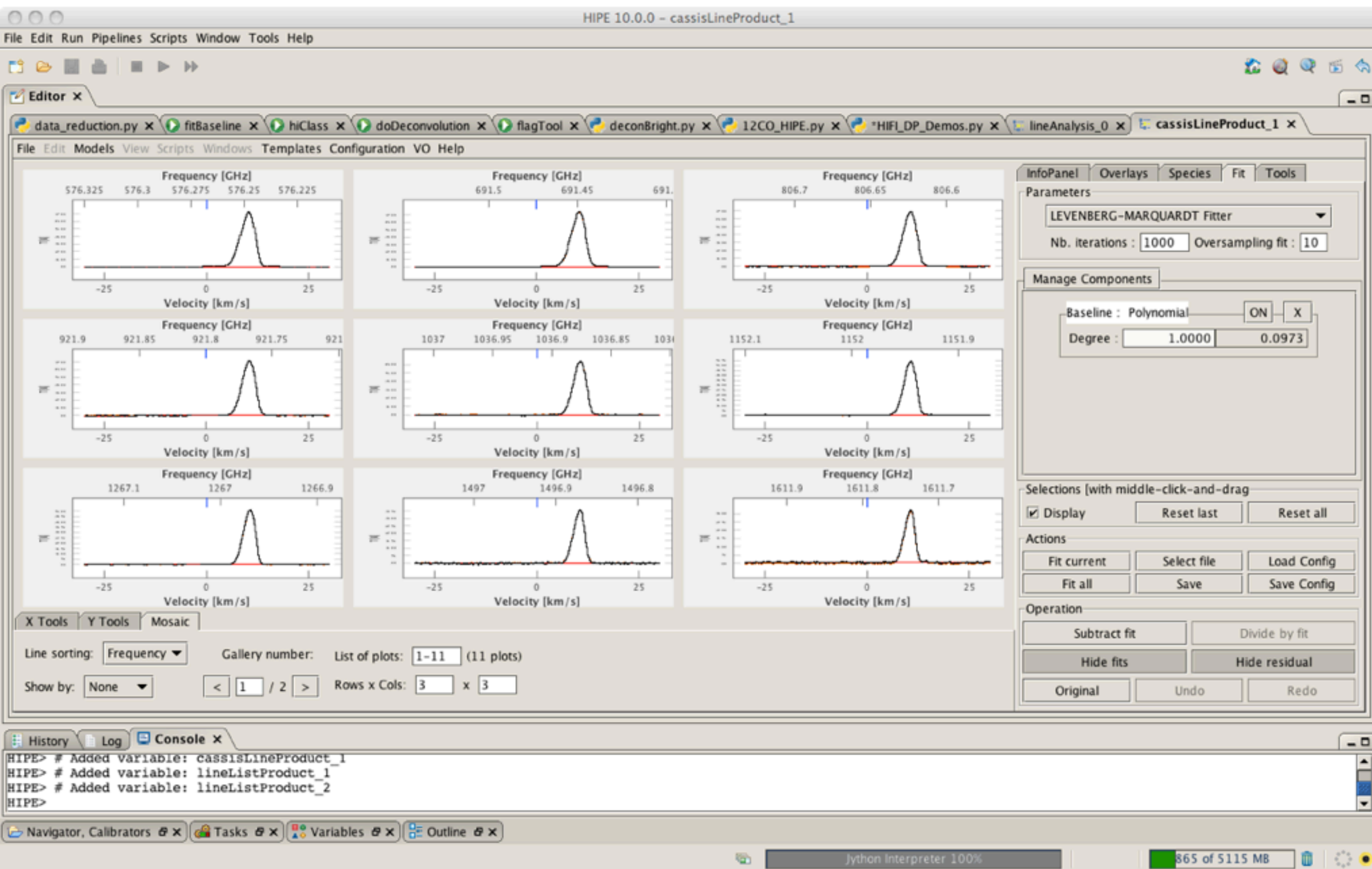
Manage Component, Add Baseline, Polynomial



1: Use your mouse (middle-click, or alt-left click if you don't have a mouse) to select the area to be fitted (here in blue). You must avoid all the lines.

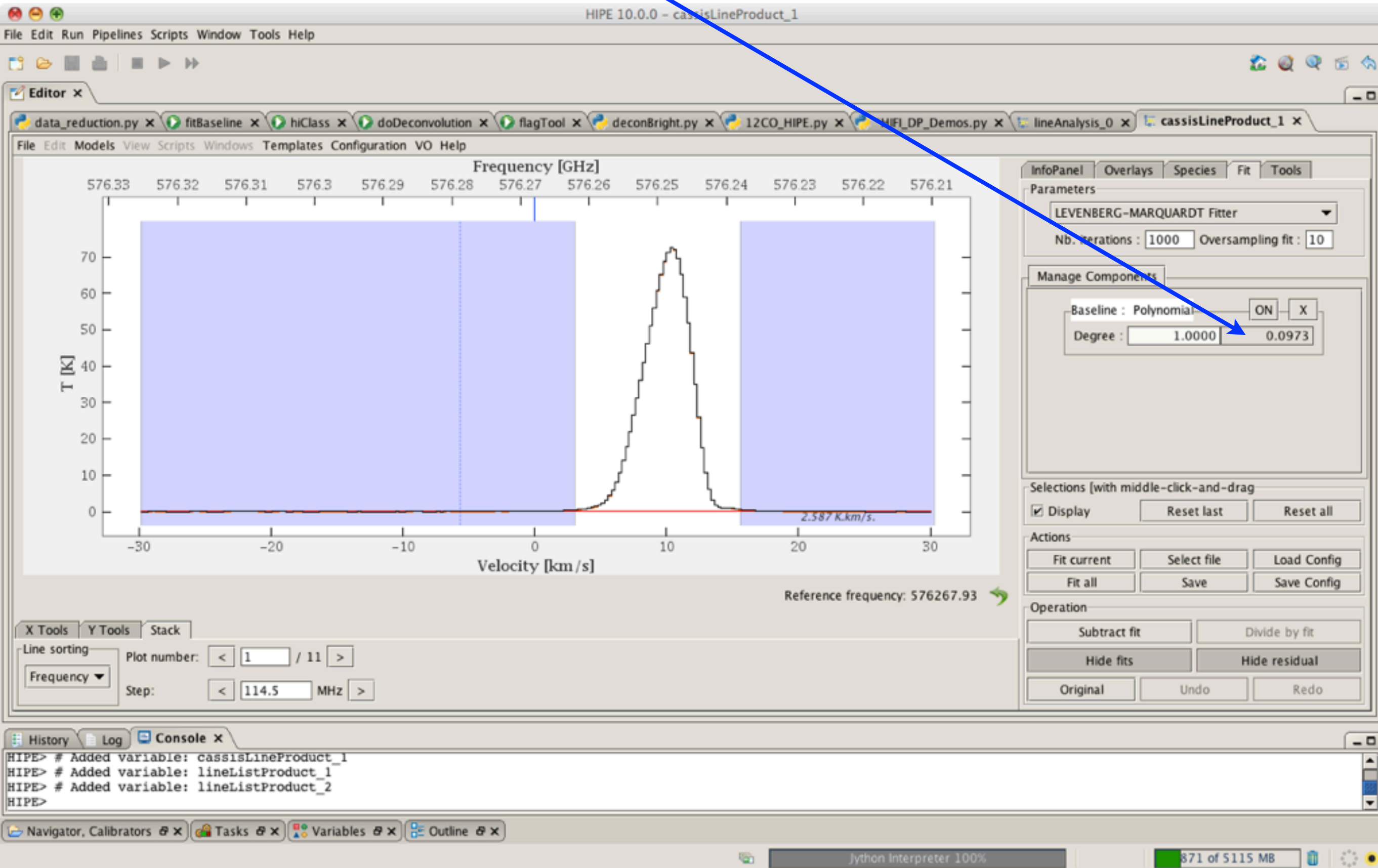


Follow the same steps as in Slides 7 and 8.



Then, double-click on each spectra and get your rms (see an example in the next slide).

Here is your rms



Once the Chi 2 modeling has finished, 3 tabs are opening

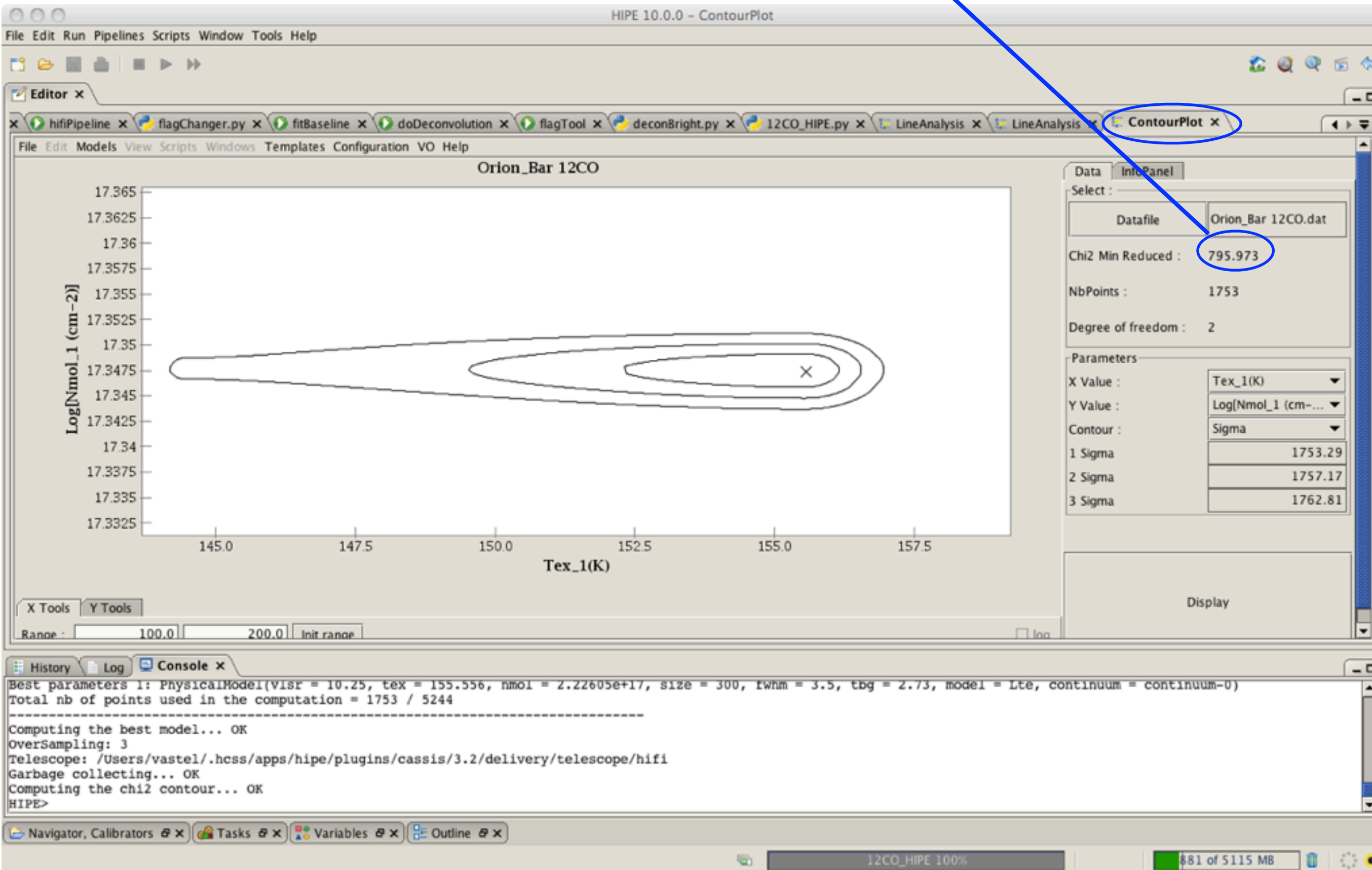
The screenshot displays the HIPE 10.0.0 software interface. The main editor window shows the `12CO_HIPE.py` script, which includes imports for `ScriptEnvironment`, `time`, `Component`, `Range`, and `UserInputs`. The script defines input parameters such as `Range.unit`, `rl`, `sourceName`, `speciesName`, `myDir`, and `myName`. It also defines user inputs for `fileName`, `suffixName`, `extensionName`, `telescope`, `tuningRange`, `tuningBand`, `aijMin`, `eup`, `kup`, `template`, `moltags`, `tmb2t`, and `isUnique`.

The console window at the bottom shows the execution results, including the best parameters for the physical model, the total number of points used, and the completion of the chi2 contour computation. The console output is as follows:

```
Best parameters 1: PhysicalModel(visr = 10.25, tex = 155.556, nmol = 2.22605e+17, size = 300, fwhm = 3.5, tbq = 2.73, model = Lte, continuum = continuum-0)
Total nb of points used in the computation = 1753 / 5244
-----
Computing the best model... OK
OverSampling: 3
Telescope: /Users/vastel/.hcss/apps/hipe/plugins/cassis/3.2/delivery/telescope/hifi
Garbage collecting... OK
Computing the chi2 contour... OK
HIPE>
```

Three tabs are open in the top right corner: `LineAnalysis`, `LineAnalysis`, and `ContourPlot`. The `ContourPlot` tab is highlighted with a blue circle. A blue arrow points from the text "Once the Chi 2 modeling has finished, 3 tabs are opening" to the `ContourPlot` tab.

Not so good but we fixed fwhm and Vlsr!!!



These are the parameters found for the best χ^2 fit, as well as the fixed parameters, that are used for the LTE modeling of the transition.

HIPE 10.0.0 - LineAnalysis

File Edit Run Pipelines Scripts Window Tools Help

Editor x

hifiPipeline x flagChanger.py x fitBaseline x doDeconvolution x flagTool x deconBright.py x 12CO_HIPE.py x **LineAnalysis** x LineAnalysis x ContourPlot x

File Edit Models View Scripts Windows Templates Configuration VO Help

Data

Load /Users/vastel/DATA/HIFI Vlsr data: 0.0 km/s in: REST Telescope ???

Tuning

Range min: 479.0 max: 1910.0 GHz Band: 60.0 km/s

Threshold

Eup min: 0.0 max: 2000.0 K Aij min: 1E-5 max: *

Jup min: * max: * Kup min: * max: * Lup min: * max: * Mup min: * max: *

LTE-RADEX ☒

Parameters

Telescope: hifi ☒ Tmb->Ta conv hifi

Noise rms: 0.0 mK

Oversampling Oversampling: 3.0

Component 1 ☒ +

Mode: Full LTE

Molecules: -- Operations --

☒ Interacting

Geometry: Slab

Tbg [K]: 2.73

N(H₂) [cm⁻²]: 7.5E22

V_{lsr}: 10.25 km/s

Continuum continuum-0

Species	Tag	Database	Compute	N(Sp) (cm ⁻²)	Abundance (/H ₂)	Tex (K)	FWHM (km/s)	Size (")
CO, v=0	28503	CDMS	<input checked="" type="checkbox"/>	2.23E17	0.001	155.556	3.50	300.00

History Log Console x

Best parameters 1: PhysicalModel(Vlsr = 10.25, tex = 155.556, nmol = 2.22605e+17, size = 300, fwhm = 3.5, tbg = 2.73, model = lte, continuum = continuum-0)

Total nb of points used in the computation = 1753 / 5244

Computing the best model... OK

Navigator, Calibrators x Tasks x Variables x Outline x

12CO_HIPE 100%

891 of 5115 MB

These are the modeled transitions, for the best χ^2 fit. Note that we fixed the beam dilution, the FWHM as well as the Vlsr.

