

Summary of CASSIS scripts

CASSIS team

April 2021

1 Summary tables

In the following, we refer to the directory “path_to_cassis/delivery/data/” as the “cassisDataPath”. All input spectral data files for the RG and MCMC scripts are in this directory.

1.1 Regular grid (RG) scripts

- Each script creates the following files :
 - myName.dat : χ^2 values
 - myName_bestModel.lis : best model
 - myName.lam : line analysis configuration for the best parameters

where myName is a variable defined at the beginning of the script and is given in the table below. By default, these files are written in the cassisDataPath (can be changed in the script).

- Each script then calls the python script “plot_chi2_RG.py” : you need to edit the jython script to specify the path to your python3 with pandas, matplotlib, numpy, and scipy installed. This script :
 - reads the myName.dat file and displays a triangle plot showing pairwise χ^2 contours.
 - generates a png file of the figure with the same name and in the same directory as the .dat file (can be changed in the python script).

N.B. : As provided, the LTE scripts take between ~ 30 and ~ 90 s, and the Radex script takes ~ 3.5 min.

Script name	What it does	Input	myName
CO_RG.py	Line analysis, LTE, two molecules (^{12}CO , ^{13}CO), one component	CO.fus	IRAS16293.12CO.13CO.1comp.lte
CO_RG_LTE.py	Line analysis, LTE, two molecules (^{12}CO , ^{13}CO), two components, same isotopic ratio	I16293.Synthetic_CO.lis	I16293.Synthetic.12CO.13CO.2comp.lte
H2CO_RG_LTE.py	Line analysis, LTE, one molecule (H_2CO), two components	Orion.Synthetic.H2CO.lis Orion.Synthetic.Continuum	Orion.S_para-H2CO.2comp.lte
H2CO_RG_Radex.py	Line analysis, Radex, one molecule (H_2CO), two components (one fixed)	Orion.Synthetic.H2CO.lis Orion.Synthetic.Continuum	Orion.S_para-H2CO.2C_radex
Inverse_PCygni.py	PCygniModel	H2CCO.lis	IRAS16293-2422_H2CCO ^a

^a Currently, the PCygniModel does not save the χ^2 values, and the only output is the best model (myName_bestModel.lis).

1.2 Monte-Carlo Markov Chain (MCMC) scripts

- Each script creates the following files :
 - myName.dat : parameter values for each run of the MCMC algorithm
 - myName_bestModel.lis : best model
 - myName.lam : line analysis configuration for the best parameters

where myName is a variable defined at the beginning of the script and is given in the table below.
By default, these files are written in the cassisDataPath (can be changed in the script).

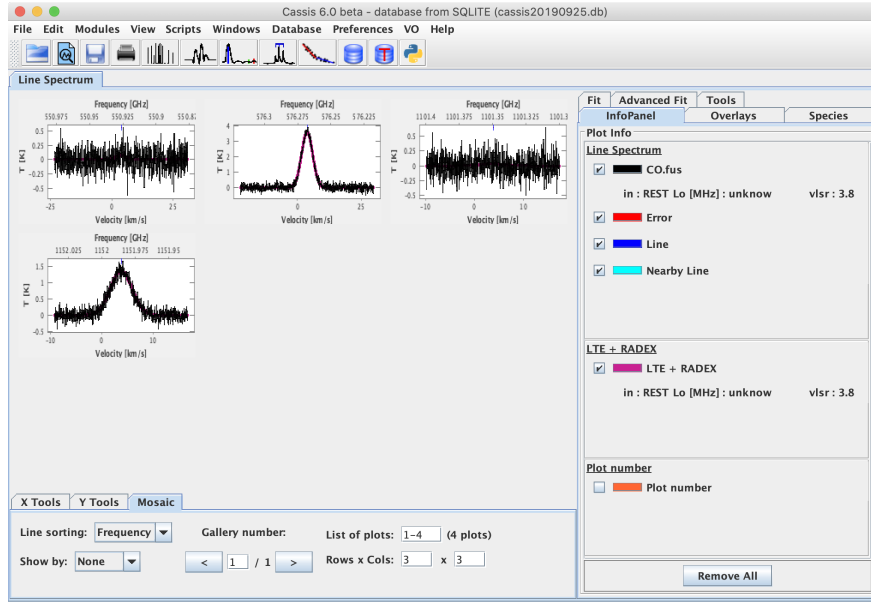
- Each script then calls the python script “Plots_MCMC.py” : you need to edit the jython script to specify the path to your python3 with pandas, matplotlib, numpy, and scipy installed. This script :
 - reads the myName.dat file and displays a corner plot showing 1-D and 2-D probability density distributions (one figure per component).
 - generates a png file of the figure(s) with the same name and in the same directory as the .dat file (can be changed in the python script).

Script name	What it does	Input	myName
CH3OH.MCMC.LTE.py	Line analysis, LTE, one molecule (CH ₃ OH), two components	Orion.Synthetic.CH3OH.lis Orion.Synthetic.Continuum	Orion.Synthetic.CH3OH.2C.lte
H2CO.MCMC.Radex.py	Line analysis, Radex, two molecules (o-H ₂ CO and p-H ₂ CO), two components	Orion.Synthetic.H2CO.lis Orion.Synthetic.Continuum	Orion.Synthetic.o-p-H2CO.2C.radex

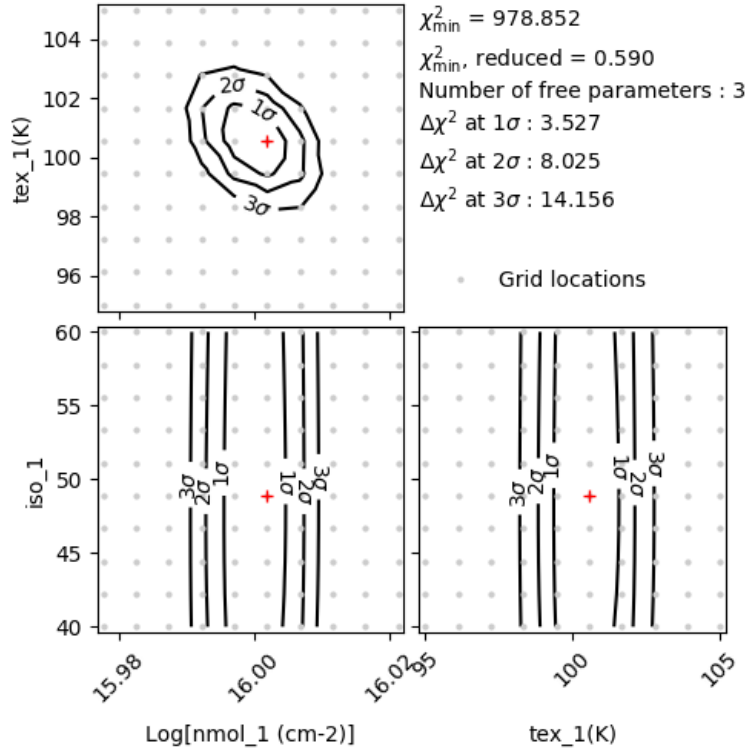
1.3 Other scripts

Script name	What it does	Remarks
Database.py	Demonstrates the use of the CASSIS database API	
ghosstRead.py	Reads GHOSST xml spectrum file and displays the spectrum in CASSIS	A GHOSST file downloaded from http://ghosst.osug.fr
inspectsFitsFile.py	Prints header information for each HDU of a given fits file	Information printed in the terminal which CASSIS was started from (not in the jython console)
rotationalpoint.py	Prints point values from Rotational Diagram	
saveSpectrumToFits.py	Creates a spectrum and saves it to a fits file	
testPartitionFunction.py	Computes the partition function of a molecule at a given temperature	
useMultiDB.py	Connects to multiple databases at the same time	

2 Example plots



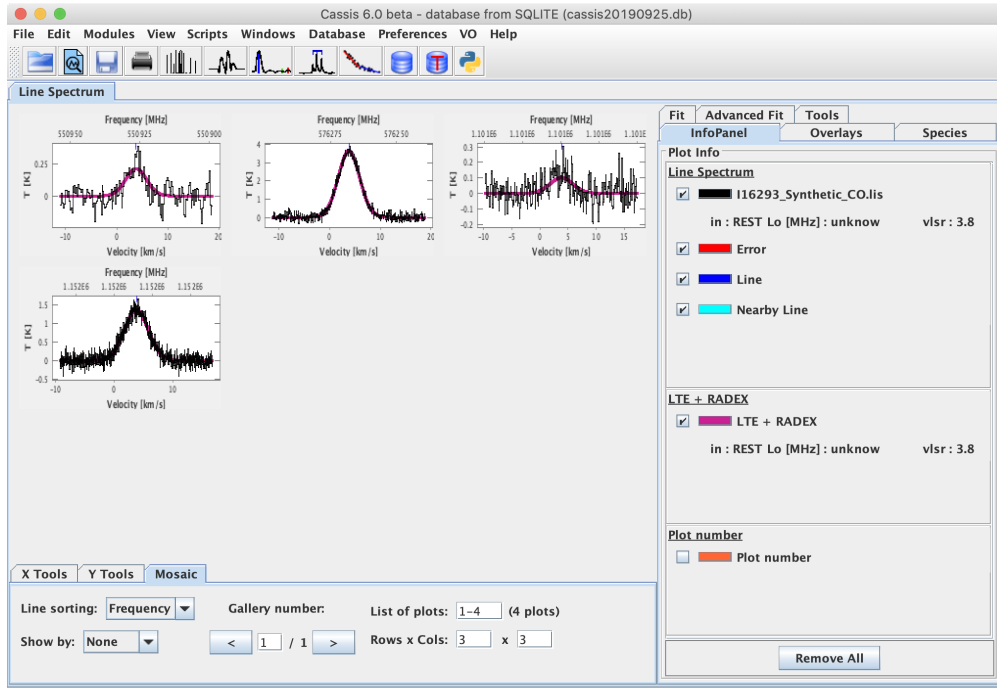
(a) CASSIS plots of the best model.



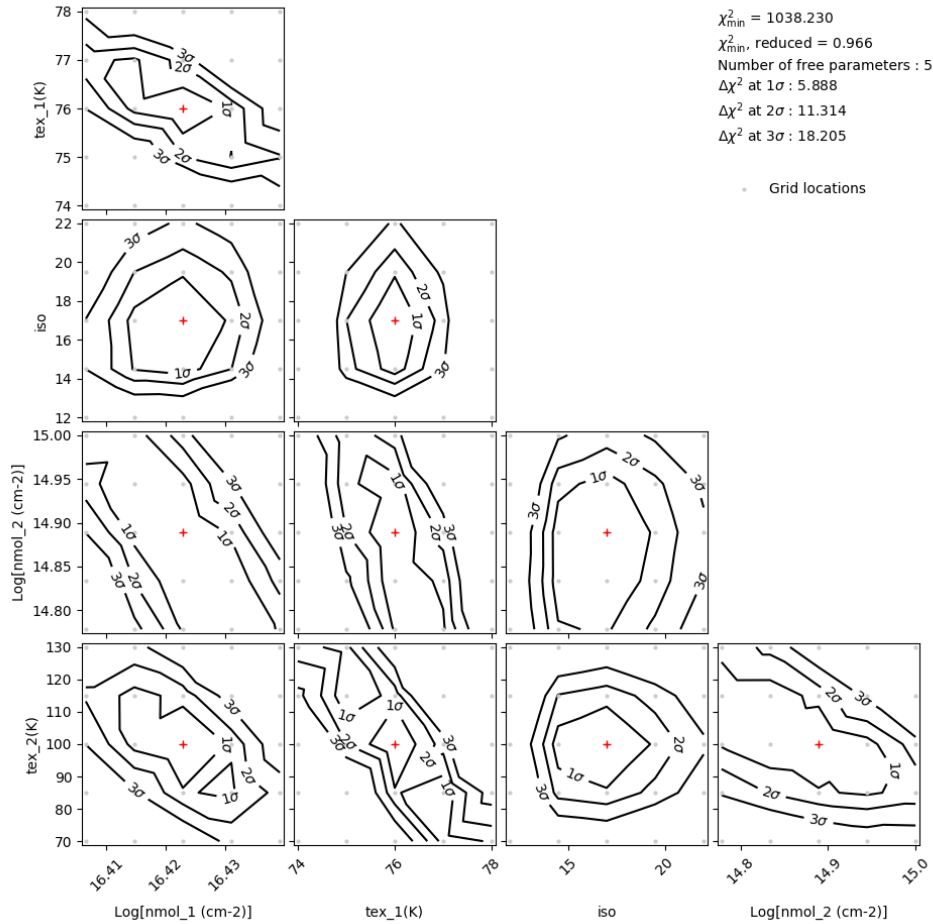
(b) χ^2 contour plots.

Figure 1: Outputs from CO_RG.py

[\[Back to the table.\]](#)



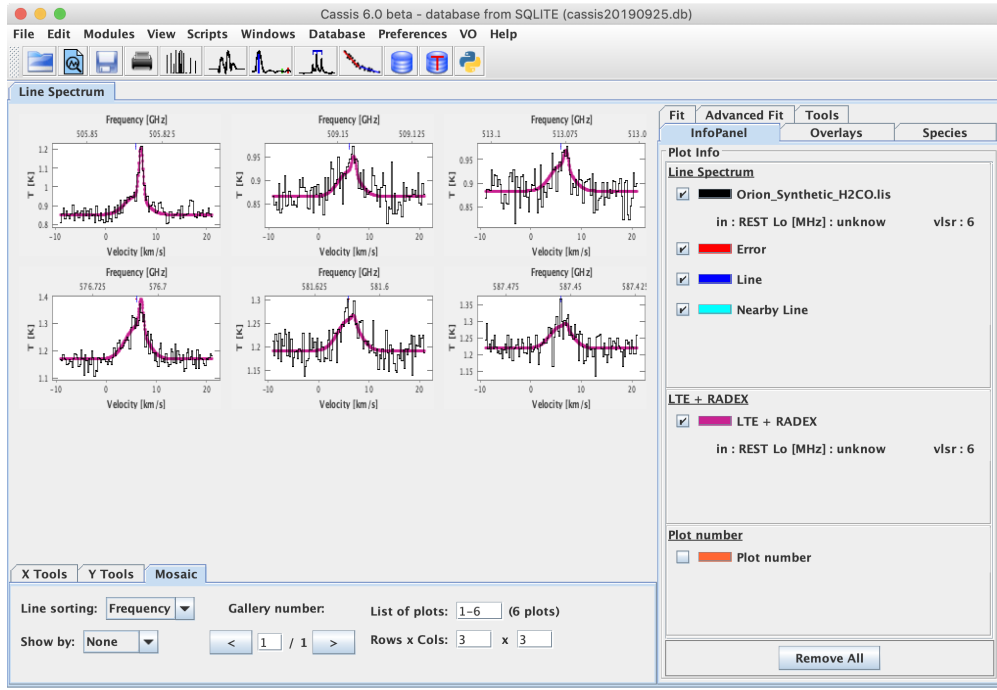
(a) CASSIS plots of the best model.



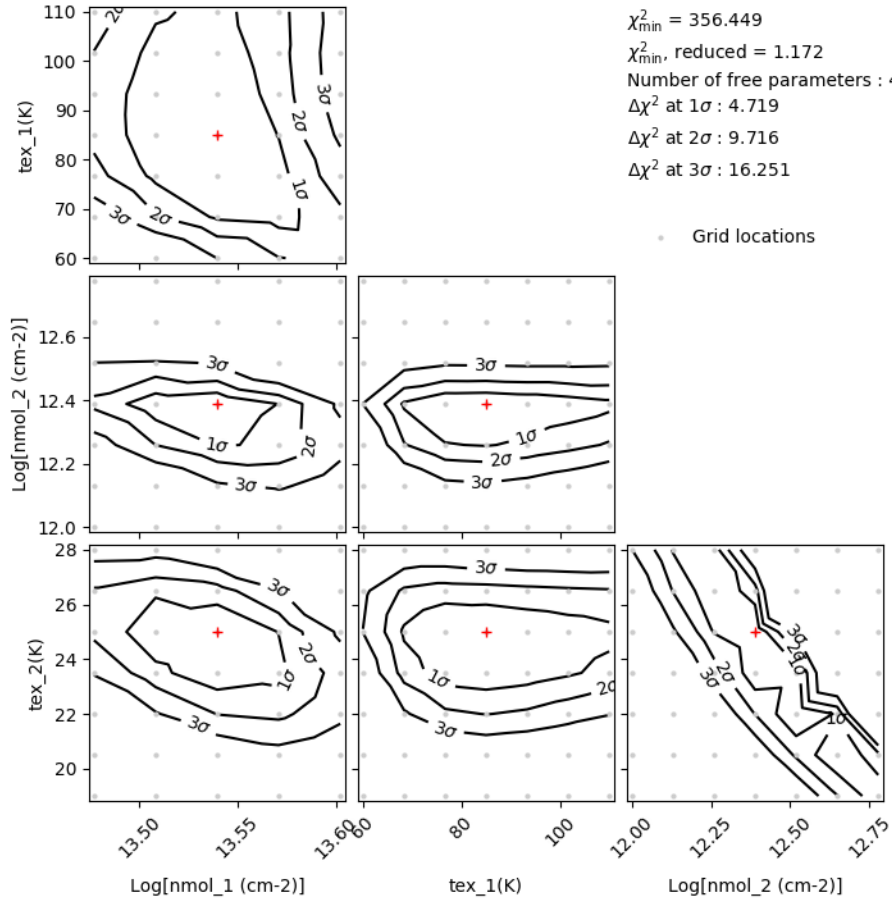
(b) χ^2 contour plots.

Figure 2: Outputs from CO-REG-LTE.py

[\[Back to the table.\]](#)



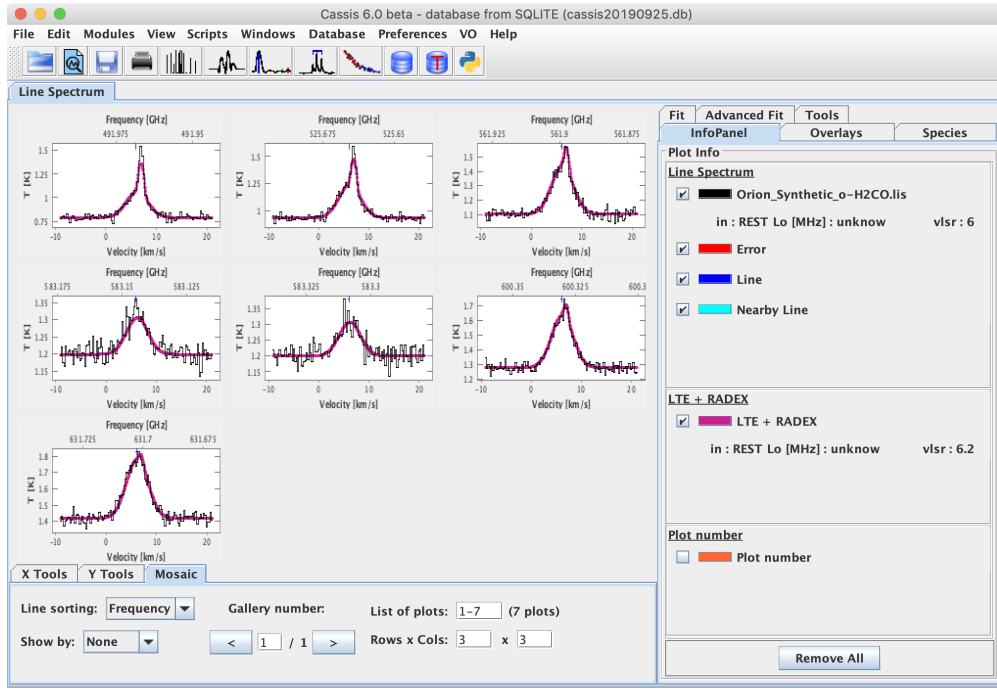
(a) CASSIS plots of the best model.



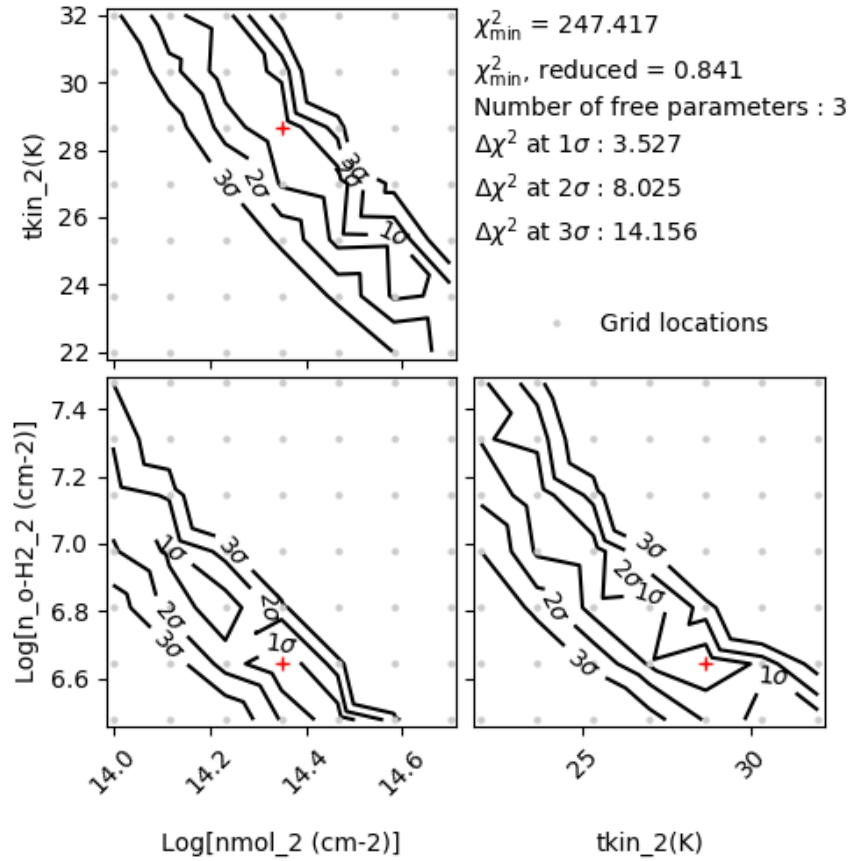
(b) χ^2 contour plots.

Figure 3: Outputs from H2CO_RG_LTE.py

[\[Back to the table.\]](#)



(a) CASSIS plots of the best model.



(b) χ^2 contour plots.

Figure 4: Outputs from H2CO_RG_Radex.py

[\[Back to the table.\]](#)

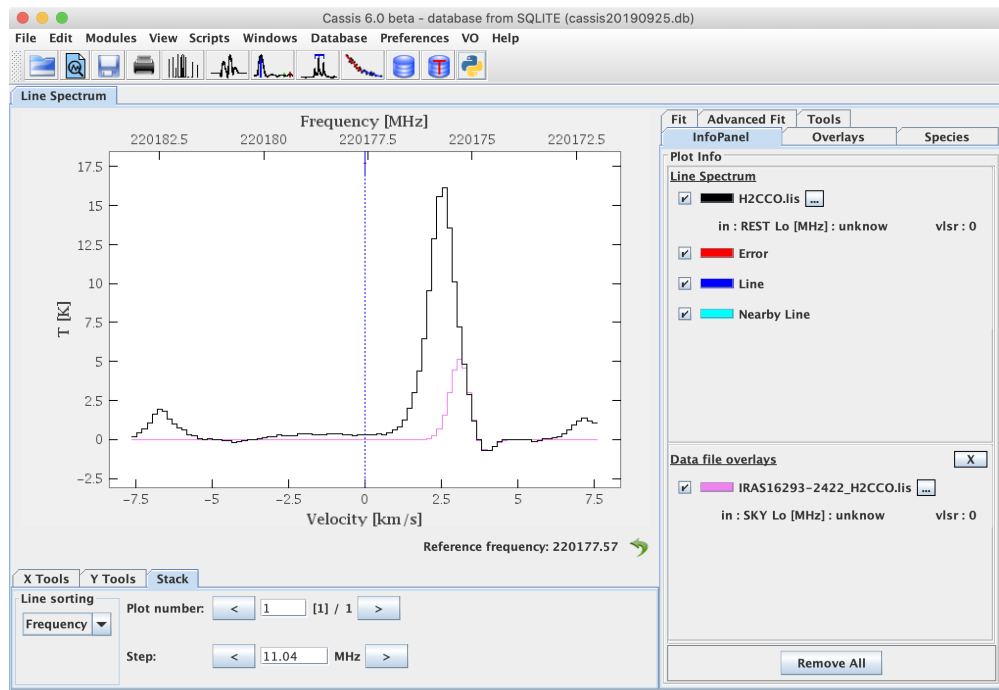
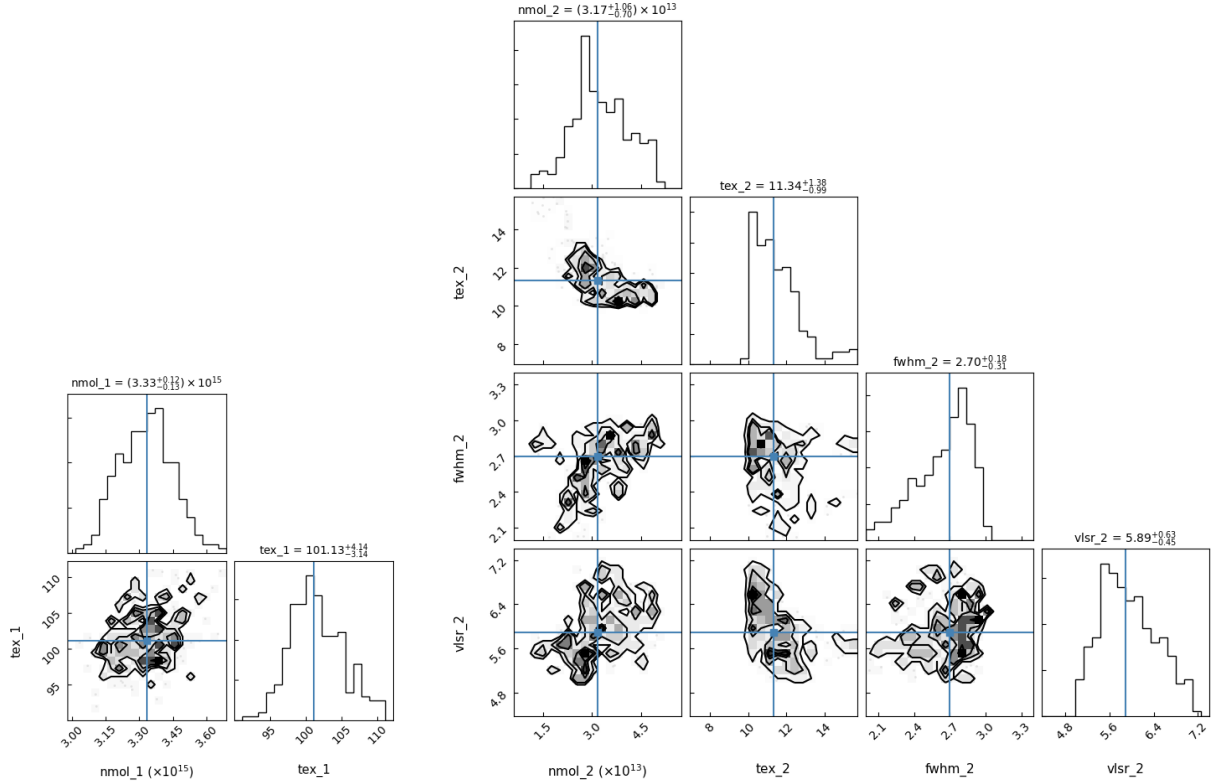


Figure 5: Output from Inverse_PCygni.py (No contour plots available)

[\[Back to the table.\]](#)



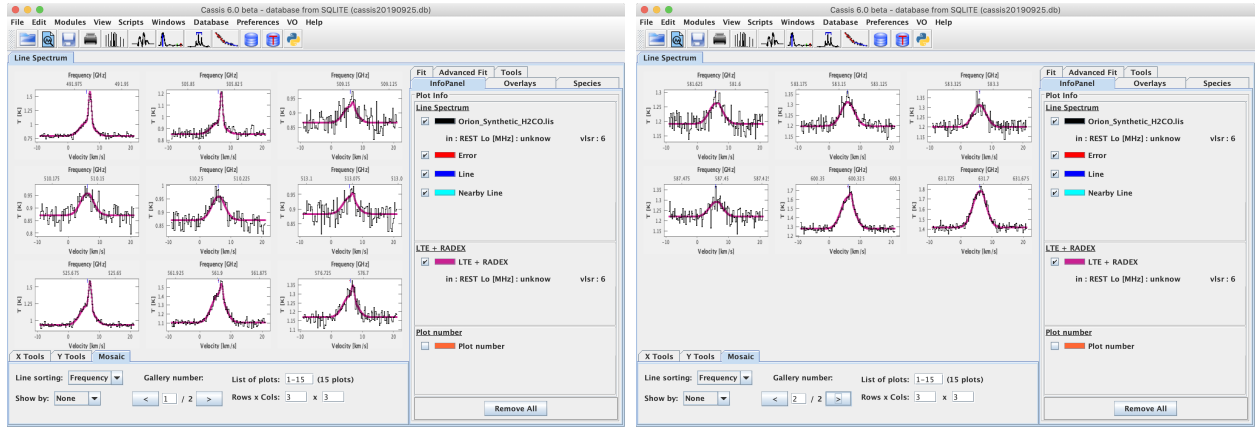
(a) CASSIS plots of the best model.



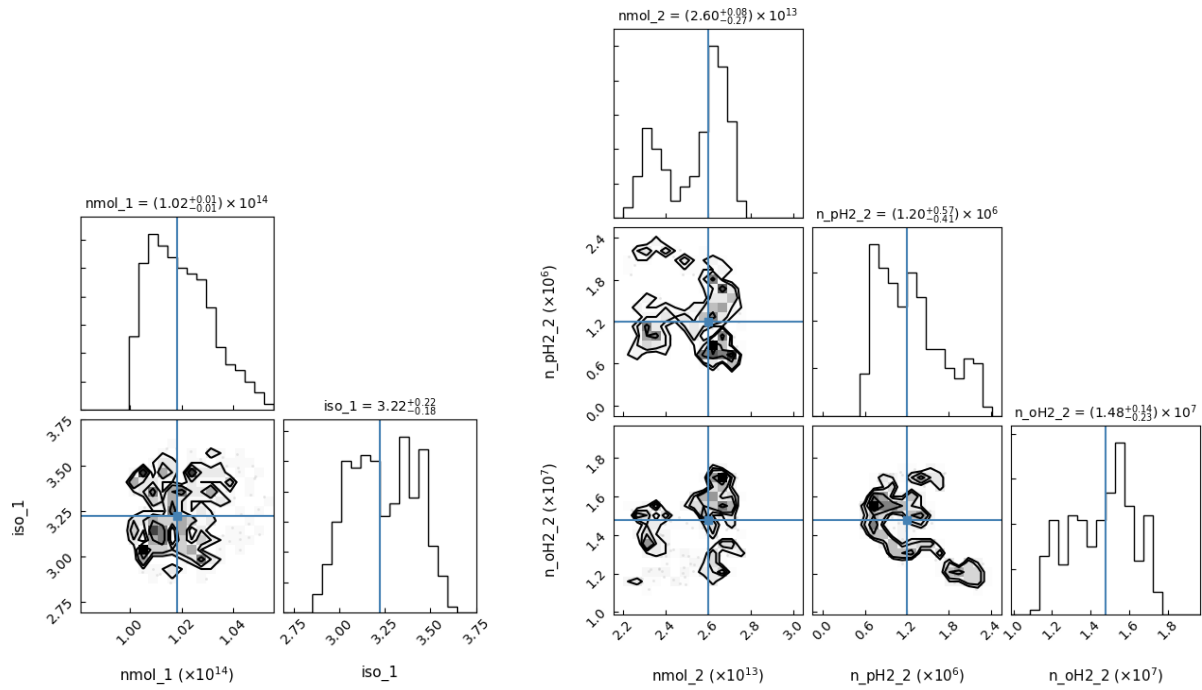
(b) Corner plots.

Figure 6: Outputs from CH3OH.MCMC.LTE.py

[\[Back to the table.\]](#)



(a) CASSIS plots of the best model.



(b) Corner plots.

Figure 7: Outputs from H2CO_MCMC_Radex.py

[\[Back to the table.\]](#)