Adaptive Radiative Transfer Innovations for Sub-millimeter Telescopes (ARTIST)

Graphical User Interface - V1.0 Beta

including

Image Viewer and ANalyzer (IVAN) - V1.0Beta

Attila Juhász
and the ARTIST team

ARTIST GUI is a graphical front-end for the ARTIST package written in python. This document gives a guided tour through the user interface how to set up a model and do some basic analysis on the model and on the calculated images.

Startup

Before we start the ARTIST GUI we must go to the ARTIST root directory and execute the command:

```
$>source artistrc.sh
```

artistrc.sh should be used for bash while for csh/tcsh we should replace artistrc.sh with artistrc.csh. These scripts set the proper environment variables for ARTIST . The ARTIST GUI can be started from the terminal with the artist command.

If the artist command is executed without any options a small dialog will pop up from which we can choose from three options: i) open an already existing model, ii) create a new model, iii) exit. We can also skip the startup dialog and open a model from the command line using the -model option followed by the name of the model:

```
$>artist -model demoModel.mdl
```

If the the model does not exists a dialog will pop up if we want to create a new model with the given name.

Main window

All input fields required to set up a model are located in the main window (see Figure 1). The input fields for the parameters are grouped in three tabs of related parameters (e.g. model input, raytracing/images, grid/excitation). At the bottom of the window there are four buttons used to launch a diagnostic window (Visualize 1D/2D/3D), the image viewer, or to run the line radiative transfer code (Lime) to calculate images from the model.

The models can be saved and later-on restored. ARTIST models are stored in a directory that has a '.mdl' extension. The model directory structure is similar to that used by the Common Astronomy Software Applications (CASA) package in the sense that each model "file" is actually a directory, containing all relevant input files (e.g. dust opacity file, molecular data file, etc) required to open and reproduce the model. The model parameters are stored in the 'model.stp' file in each '.mdl' model directory.

ARTIST also uses a log file 'artist.log' to log currently most of the key steps that happen behind the windows and widgets. The artist.log file is created at the directory where we start artist. If a log file is not needed or its size is too big we can simply delete it.

Model input tab

Selection of the physical model description

The model input tab in the main window contains eight input fields, one for the model description and seven for each physical variable. ARTIST has a built in library for physical model descriptions that be accessed with the "Browse models..." button at the top of the main window. This will bring up a dialog (see Figure 3) where we can select any of the available model descriptions. The dialog also shows the bibliographic reference and lists the input parameters with a short description. After a model has been selected, the model input tab in the main window will be updated and will display the input parameters for the selected model description. The unit of the parameter is shown in brackets after the name of the parameter.

Definition of the physical variables

To calculate images/channel maps for a given model Lime requires seven physical variables; density, gas temperature, dust temperature, velocity, intrinsic line width (Doppler b parameter), magnetic field and molecular abundance. For the description of these variables we refer to the Lime User Manual. The physical variables in the model can be defined in three ways: i) given by a physical model description (e.g. Shu 1977) ii) given by a simple analytical function (e.g. power-law) iii) read from a file (ARTIST Table format). The definition type (model, analytic function, tabulated) can be selected from a drop down list next to the name of the variable. If a variable is provided by the model the default definition type is 'model'. Usually the model description does not specify all seven variables required by Lime . Variables not provided by the model description are highlighted with red. For these variables we must make an assumption and select an analytical function or grab the variable from a table file.

For the abundances another definition method, 'Regions', is present. With this selection we can define up to five regions in the model space with different (constant) abundances in each regions. The regions can be set up on the basis of spherical radius or gas temperature (e.g. to simulate freeze-out of the molecules on the dust grains).

Excitation/Grid Tab

This tab contains the parameters for Lime that are related to excitation calculation and the spatial grid. For the description of these parameters we again refer to the Lime User Manual.

Raytracing/Images Tab

This tab contains the parameters for Lime that are related to the calculation of the images/channel maps. For the description of these parameters we again refer to the Lime User Manual.

The upper half of this tab, the 'Image parameters' field, contains parameters to calculate one single image. If we wished to calculate channel maps for a model in two different transitions we need to run Lime twice, for each transitions. This is however computationally not efficient, since the only the raytracing should be done twice, the spatial grid and level population can be calculated once and used for both transitions. `Lime` / `ARTIST` supports the calculation of multiple images in a single run (i.e. after setting up the model grid and the calculating the level population raytracing/image calculation is done in multiple times). To calculate more than one image in a single `Lime` run we need to click on the 'Multiple images' check box in the middle of the tab to enable the calculation of multiple images.

In the 'Multiple image' block we can select up to four parameters to change from image to image. E.g. if we wish to calculate the channel maps in the CO J=1-0, J=2-1, J=3-2, and J=6-5 transitions we need to click on the 'Transition' check box in the 'Multiple images' field and give the indices of the four transitions in the molecular data file (e.g. by clicking on the Browse datafile... button). This way we calculate four images with the exact same parameters except for the transition that will be different in each images. For the inclination, image and velocity resolution we can set values in two ways. We can either give a comma separated list of arbitrary values or set up a regular (linear or logarithmic) grid. Obviously we need to give a different file name for each output image. The file names are generated semi-automatically. The image name begins with the word 'image' which is followed by tags and indices for each variable that changes in the images series. The tag indicates which variable changed in that image series (e.g. 'inc'-inclination, 'trs'-transition) and the index shows the place of the image in the series. We can specify the tags and their separators at the bottom of the 'Multiple images' section.

Visualize 1D/2D window

Once we set up all the model parameters in the main window we can inspect the physical variables in our model in the Visualize 1D/2D window. This window contains two tabs, one for 1D line graphs and one for 2D contour plots. The all figures are displayed using the `matplotlib` library. Each figure in `ARTIST` contains a standard `matplotlib` toolbar with eight icons (for e.g. zoom/pan, save figure). Each icon displays a brief tooltip help message when the cursor is moved over. We refer to the matploblib documentation for the detailed description of the icons
(http://matplotlib.org/users/navigation_toolbar.html).

IMPORTANT NOTE: The plots shown in this window are displayed using a cartesian grid, which is different than the 3D Delaunay grid used in the actual computation in `Lime` . The 3D Delaunay grid is calculated on-the-fly in Lime, meaning that visualization is possible only *after* Lime has been run (see 3D Visualization section). Since we wish to inspect the model structure *before* we run Lime to see if the model is indeed set up properly, we need to use another spatial grid. The plots in this window use the actual model (i.e. the definition of all physical variables, specified in the main window) but calculate the displayed variables in a cartesian grid. The parameters of the displayed grid can be set in the 'Plotted Grid' field in the 'Plot 2D tab' and in the 'Nr of points' field in the 'Plot 1D' tab.

Plot 1D

In this tab (see Figure 5) we can make 1D graphs for the physical variables along one of the coordinate axes. We can select spherical or cartesian coordinate system and the axes along which the variables should be plotted. I.e. we can plot variables versus spherical r or cartesian x coordinate. Since all models are described in 3D, for a 1D graph we should also specify the value of the two coordinate that should be kept fixed.

Plot 2D

In this tab (see Figure 5) we can display 2D contour plots by fixing one of the 3D coordinates. We can display one of the physical variables as filled contour (color) maps and overlay line contours of any of the physical variables. In the contour line field each variable has a check box to switch on/off its contour lines in the figure. Next to the variable name the extent (min, max) of the values are displayed. The details of the contour lines (e.g. number and values of the contour levels, colors/color maps of the contour lines, etc) can be set in a dialog brought up by the 'Contour grid...' buttons (see Figure 7). In the Grid Dialog we can also adjust the contour levels to the data extent by clicking on the 'Fit to data' button.

3D Visualization

The 3D visualization of the model structure defined on the 3D Delaunay grid used by `Lime` is done with an external software called Paraview (http://www.paraview.org/). Paraview is not linked to `ARTIST` , it is started and run in the background with the generated grid file (by default 'grid.vtk') as an input. The command for Paraview can be set in the Main window 'Settings' menu.

Image Viewer and ANalyzer (IVAN)

`ARTIST` comes with an image viewer, called `IVAN` . It displays channel maps, moment maps and position-velocity diagrams[1]. `IVAN` can be accessed either with the 'Image Viewer' button at the bottom of the the Main window or can also be started from the command line with the command `ivan` . We can open a file from the command line simply adding the name of the file after the command `ivan` or if `IVAN` is already running with the 'Open file' menu item or the 'Browse button' in the channel map tab. `IVAN` has three tabs in its window according to the three types of plots one can make/calculate; channel map, moment map, position-velocity diagram.

Channel maps

There are three control fields for the displayed channel map. There is a toolbar just below the figure with eight icons, which is a standard matplotlib toolbar and it is the same as that in the visualize 1D/2D window.

Stepping through the image cube

Below the toolbar we can find a scrollbar that can be used to step through the channels in the image cube. Below the scrollbar the channel index and the (radio) velocity are displayed. We can move to a specific channel with either the scrollbar or entering the index number in the 'Channel index' input field and pressing the Enter key. If the image cube has polarization axis (indicated by one of the CTYPE keywords in the fits header) a drop-down list will be visible in the right input field with a 'Stokes Axis :' label from which we can select the stokes axis index in the image cube.

[1] Note, the moment maps and p-v diagram tabs will be switched off if a 2D image without a spectral axis is opened.

Gaussian beam convolution

IVAN can convolve the image with a gaussian beam whose minor-, major-axes and the position angle should be set in the input field on the right in the Channel Map tab. The 'Enabled' check box controls whether the beam-convolved image or the original image/channel map should be displayed. Note, the convolution is done on the whole image cube at once, thus it may take some time. However, the convolved image cube is stored and no further calculation will be necessary unless we change the beam parameters. Once the convolution of the cube is done, rendering the images when scrolling through the channels in the convolved cube faster than doing on-the-fly convolution.

Image clips

The image can be clipped at any values by setting the 'Clip min.' and 'Clip max.' fields. The 'Reset' button next to the input fields will re-set the clip values to the lowest/highest values in the image cube.

Contour lines

The actual channel map as well as any of the moment maps can be overlaid with contour lines. The 'Contour lines' check box controls whether or not the contour line overlay is displayed. The details of the contour plot (e.g. number and values of the contour lines, colors/colormaps, etc.) can be set in a dialog by clicking on the 'Grid Parameters' button.

Plot spectrum

If the 'Plot spectrum' check box is checked in we can click on any point in the displayed image/channel map and a new window will appear with the spectrum at that specific spatial coordinate (i.e. pixel). If the spectrum window is already up we can click again on any point on the image and the spectrum will be updated accordingly.

Moment maps

Thresholds

One can select an interval in image/pixel values and channel indices to be used in the moment map calculations. Pixels / channels outside of the interval will be masked.

Gaussian beam convolution

See the Gaussian beam convolution section in case of channel maps.

Image clips

See the Image clips section in case of channel maps.

Contour lines

See the Contour lines section in case of channel maps.

Position-Velocity diagram

The left part of the P-V Diagram tab is similar to that of the channel map tab, a large figure displays the P-V diagram at a given position and a scrollbar below the figure can be used to step through the spatial indices. Similar to the channel maps tab we can jump to any of the indices by entering the number in the 'Position index' field and hitting the Enter key. The position angle at which the P-V diagram is calculated is displayed to the right of the 'Position index'. Similar to the convolution with a Gaussian beam , `IVAN` rotation of the whole image cube is done at once with the given position angle and the rotated cube is stored. Re-calculation of the rotated image cube is necessary only if the position angle is changed.

In the top right corner of the Position-Velocity tab a small figure is located showing one of the moment maps. A white line shows the position and direction of the cut in the image plane at which the actual P-V diagram in the large figure is shown.

Gaussian beam convolution

See the Gaussian beam convolution section in case of channel maps.

Image clips

See the Image clips section in case of channel maps.

Keyboard shortcuts

`ARTIST GUI` currently supports only the following basic keyboard shortcuts;

Linux/Unix	Mac OS X	Action
Ctrl+n	Option+n	New model
Ctrl+o	Option+o	Open model/image
Ctrl+s	Option+s	Save model
Ctrl+p	Option+p	Settings/preferences
Ctrl+q	Option+q	Quit
Ctrl+left	Option+left	Move one tab to the left
Ctrl+right	Option+right	Move one tab to the right

IMPORTANT NOTE: To calculate a P-V diagram we should also have SciPy installed on the the system. In the lack of SciPy the P-V diagram tab will not be displayed. The rest of `IVAN` does not use SciPy and expected to work with full functionality without SciPy.

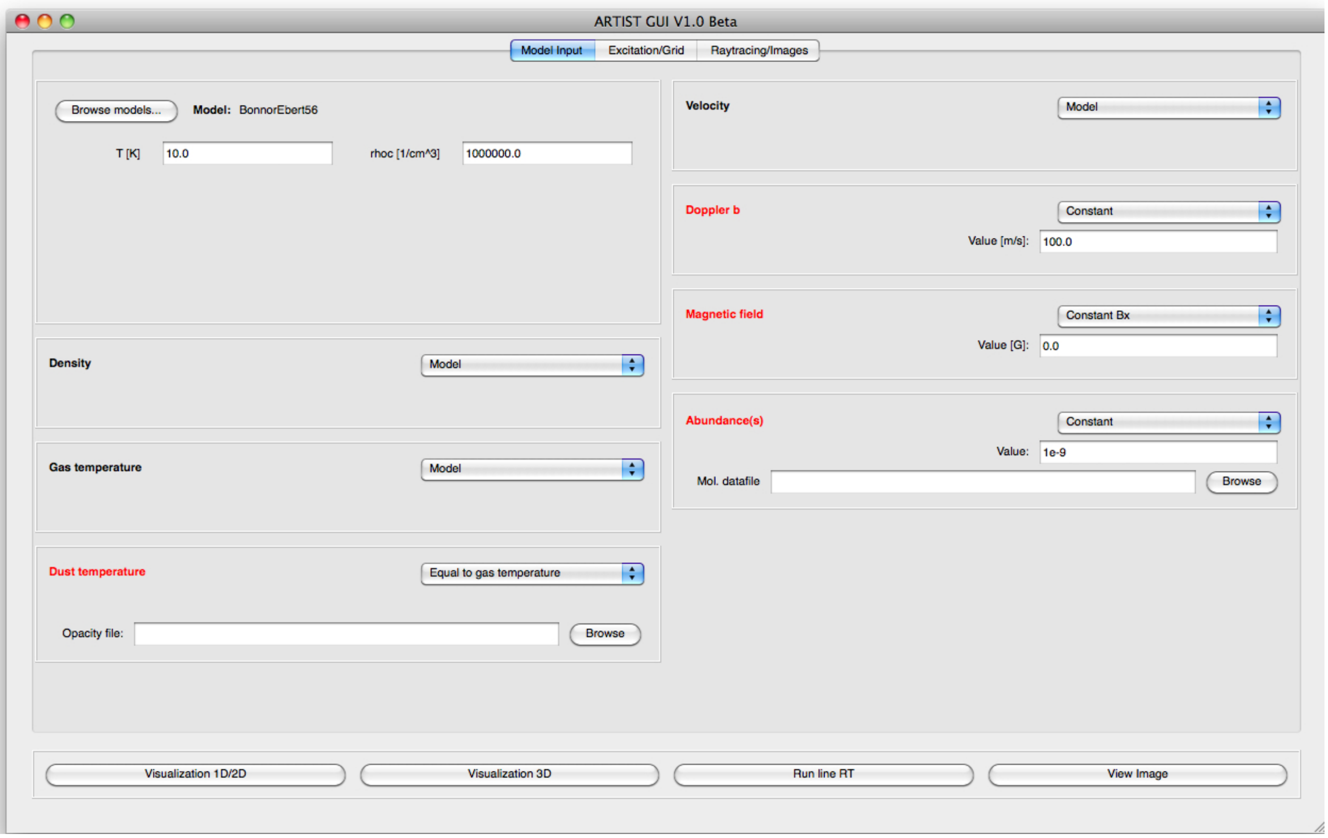

Figure 1: ARTIST Main control window

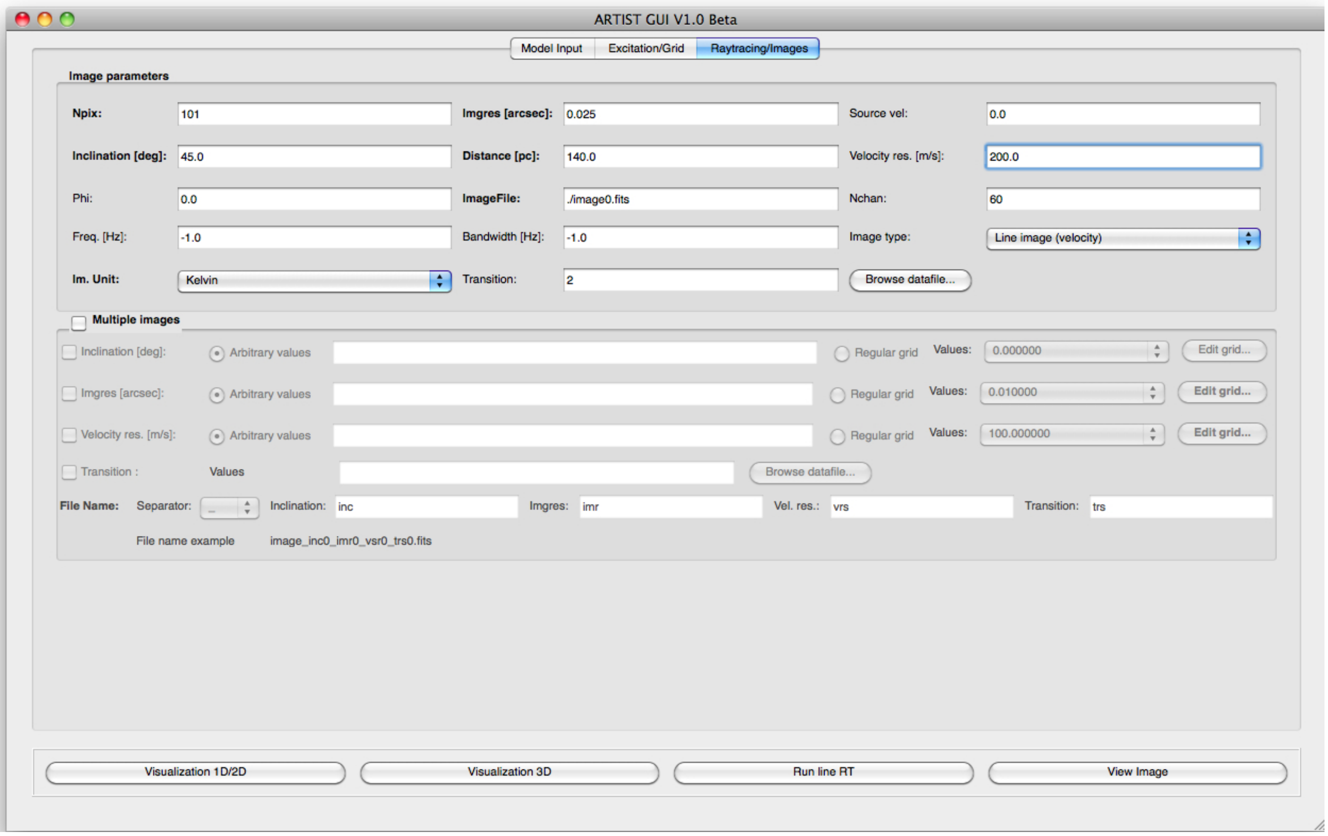

Figure 2: `ARTIST` Main control window

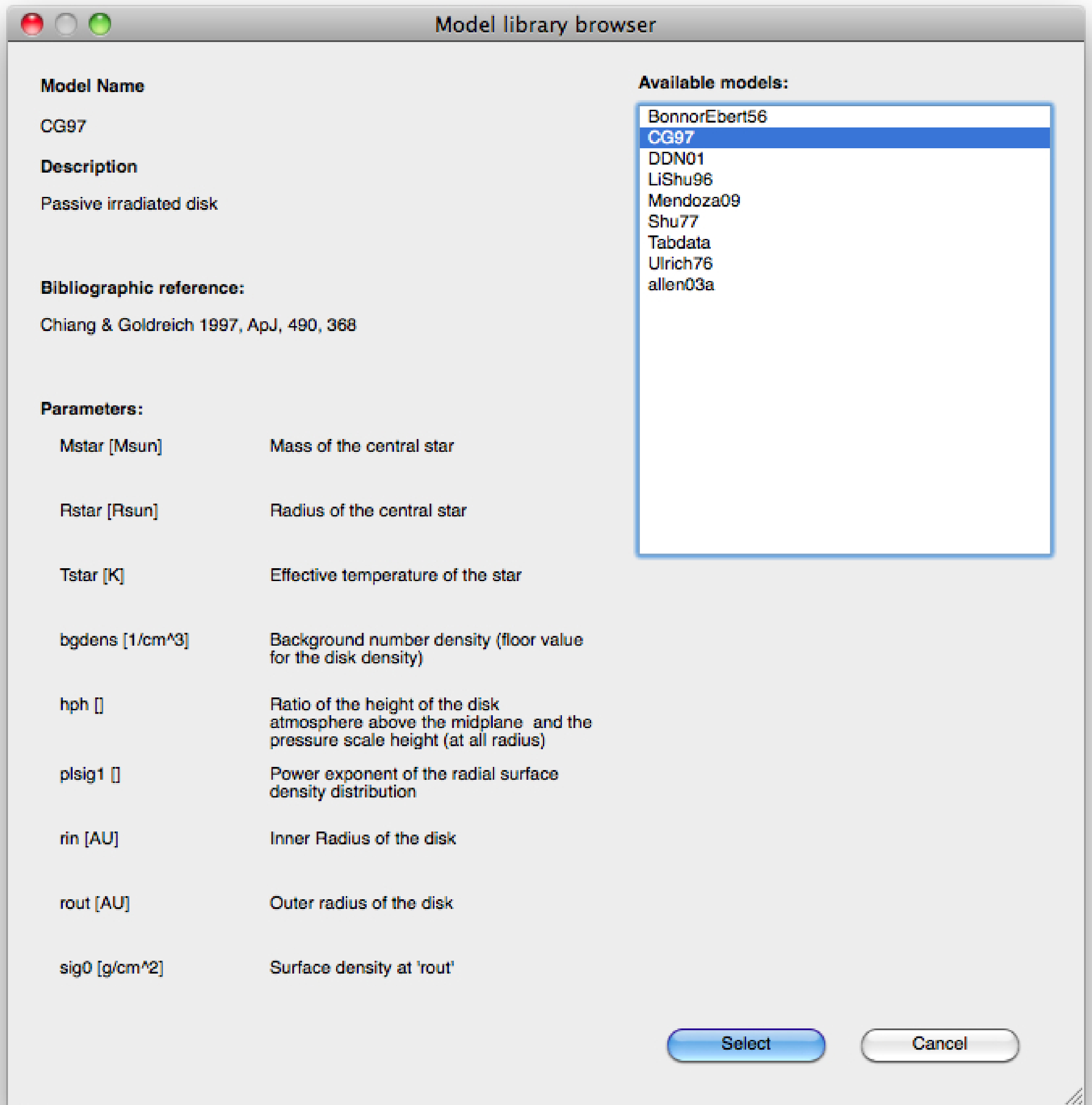

Figure 3: `ARTIST ModelLib` browser dialog

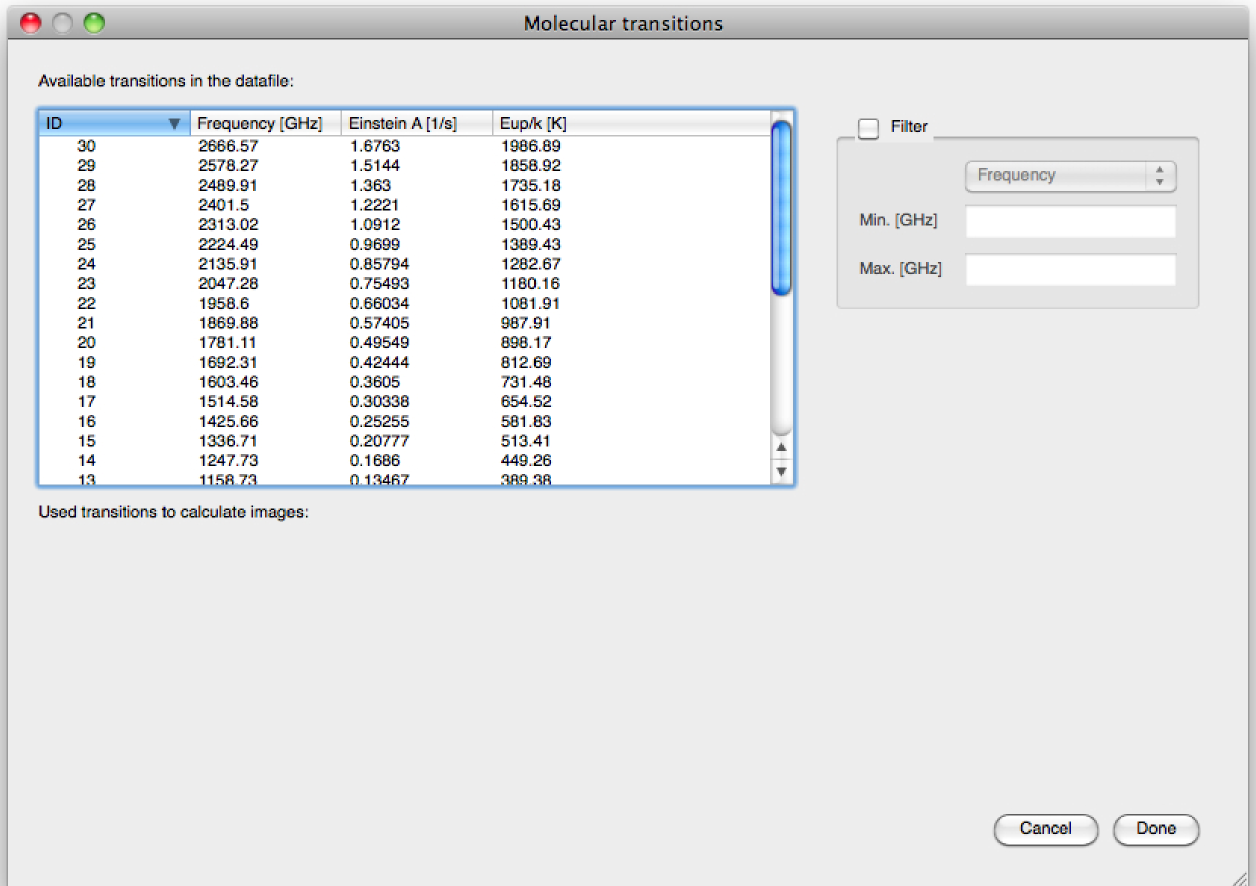

Figure 4: `ARTIST` molecular data file browser

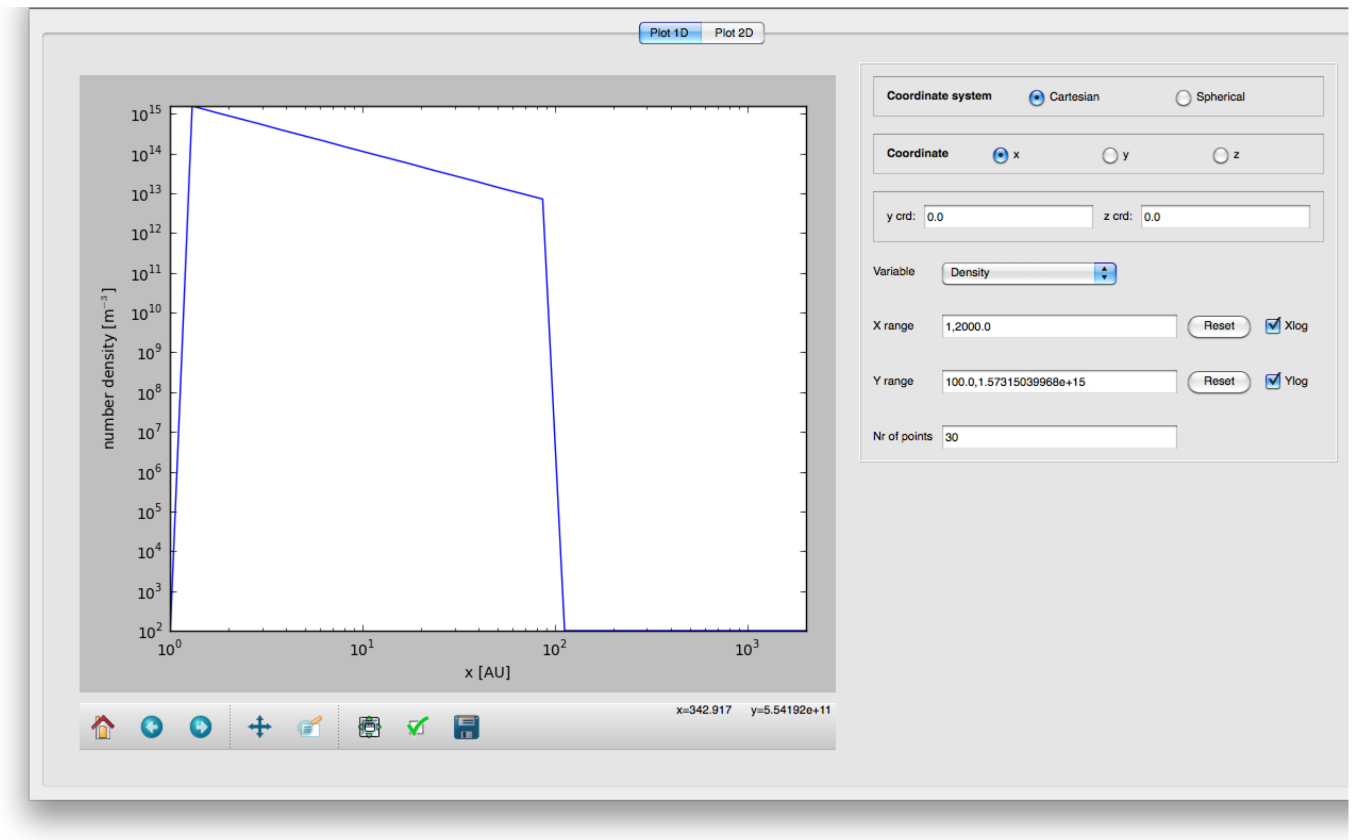

Figure 5: 1D Plots window

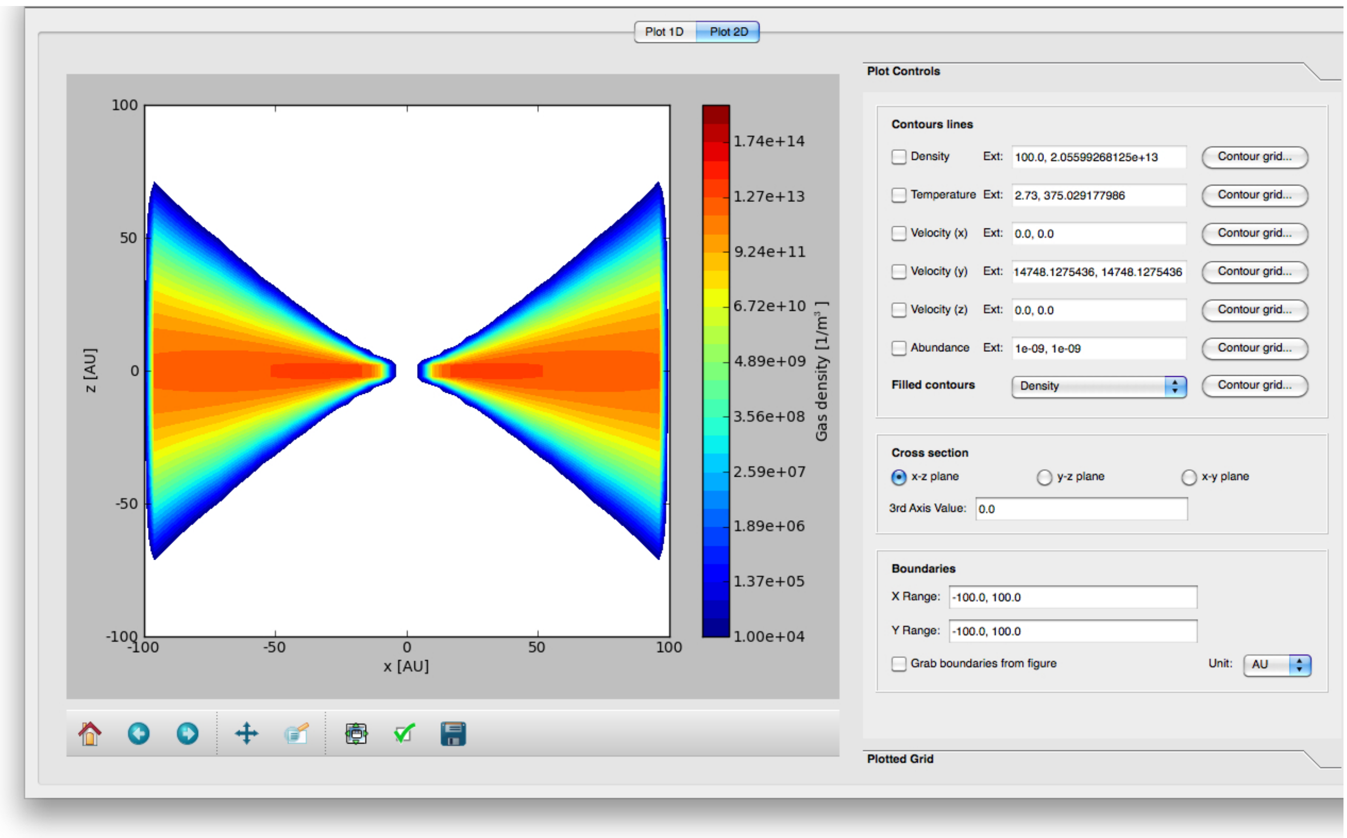

Figure 6: 2D Plots window

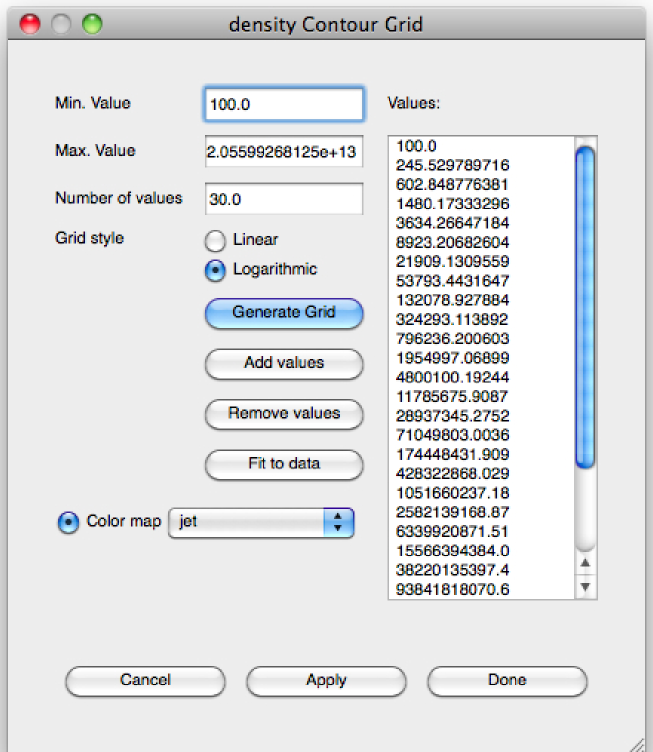

Figure 7: Grid dialog

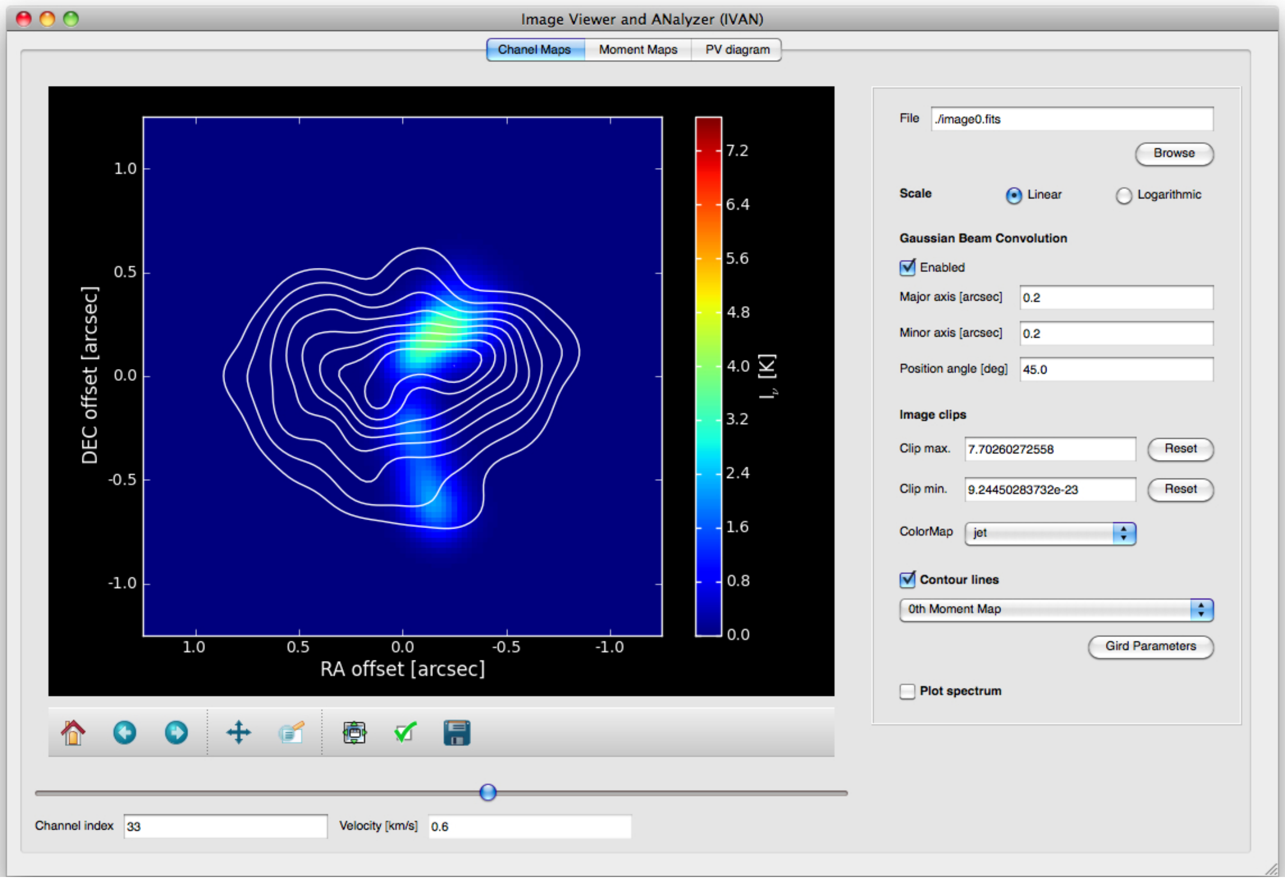

Figure 8: IVAN channel map tab

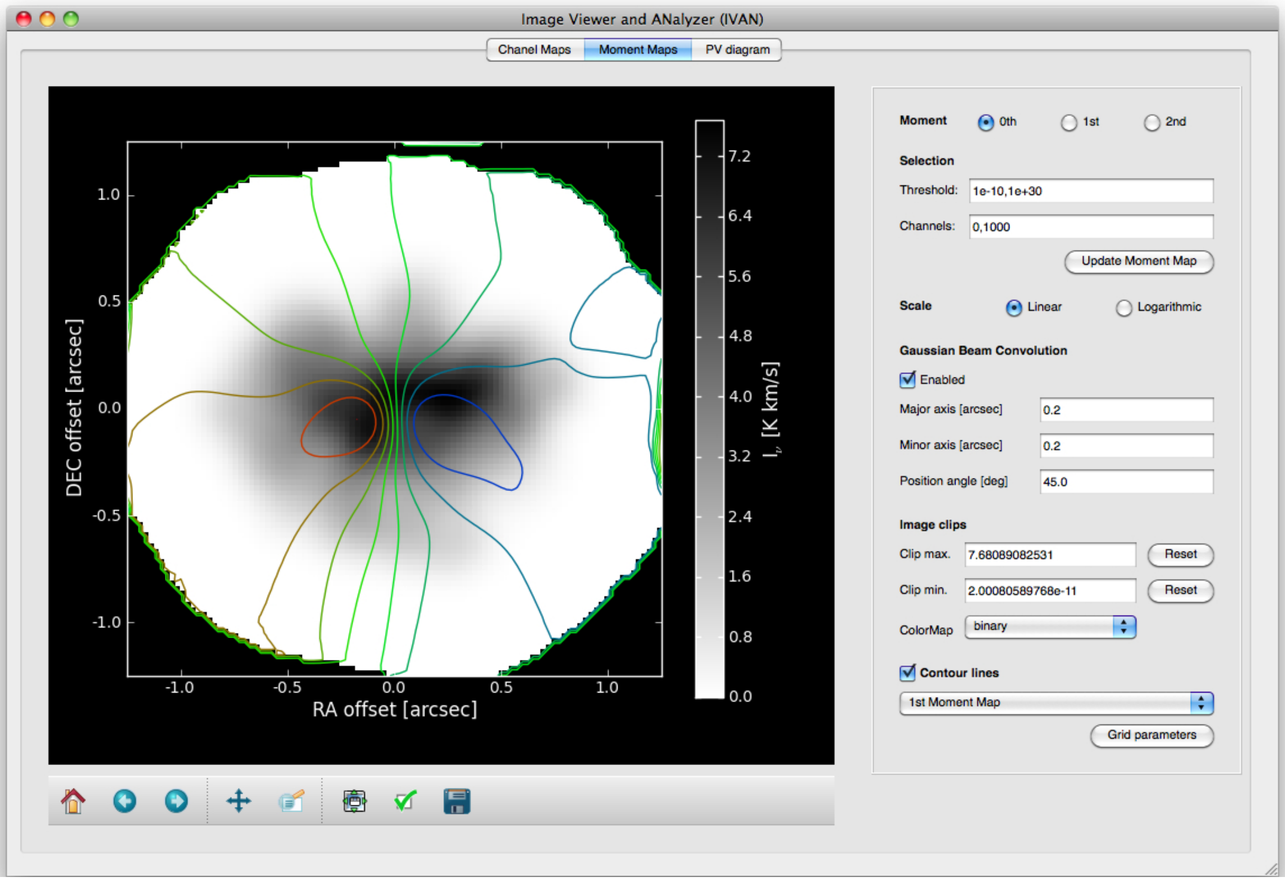

Figure 9: `IVAN` moment map tab

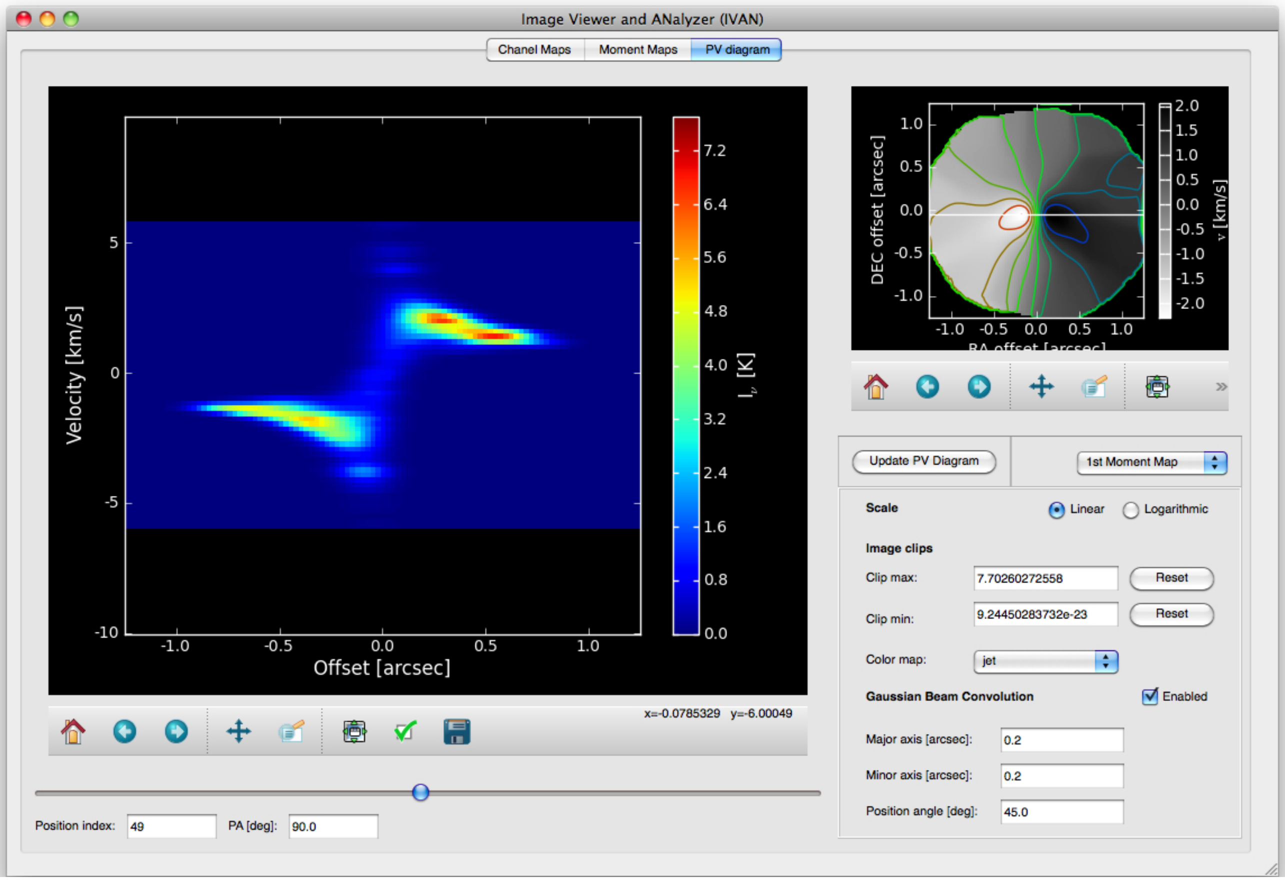

Figure 10: `IVAN` P-V diagram tab