DOCUMENTATION

IDAT : the ISM Data Analysis Toolbox

IDAT version 2.1.4

August 17, 2021

1 Overview

IDAT, the ISM Data Analysis Toolbox, provides a suite of tools to analyze and visualize the data produced by the codes of the ISM Services Platform (http://ism.obspm.fr). It is usable both on data produced after downloading and running the codes locally, and on data downloaded from ISMDB[1].

IDAT allows to:

- extract some desired computed quantities into text files
- produce graphs allowing to visualize the spatial profile of the data

IDAT includes a graphical interface but can also be used in command line to quickly extract and produce plots on several models.

✓ **More universality with IDAT 2.1.4.** IDAT 2.1.4 is an updated version of IDAT that is compatible with more versions of HDF5 files produced by the codes on the ISM platform.
Previously, depending on the version of the codes, one had to use dedicated versions of IDAT. Starting at IDAT version 2.1.4, the analysis tools is more universal. This requires to install the new version of ismpy provided with IDAT 2.1.4.

Note: In this documentation, reference is often made to the Meudon PDR code as an example of applications. IDAT can be used in the same way on any ISMServices code producing HDF5 output files, such as the Paris-Durham shock code.

[1] ISMDB: http://ismdb.obspm.fr

2 Installation and ismpy

Some python packages are required to use IDAT. To ease the use of IDAT in various operating systems and configuration, we provide a virtual python environment named *ismpy*. Thanks to it, IDAT should run properly on various operating systems and configurations.

When you first run a new version of IDAT, it is strongly recommended to de-install any previous version of ismpy and install the new one.

The only dependencies that need to be installed beforehand are :

- Python 3 with the development libraries and headers (often named `python-dev` instead of `python` in the operating system package managers), which you can install through your usual package manager (e.g. `apt-get` for Debian-based Linux distributions, ...).
- The virtual env (`venv`) module for Python 3 if it's not installed by default with python (e.g. `apt-get install python3-venv`).
- `sqlite3` with the development libraries and headers (often named `sqlite3-dev` in the operating system package managers). You can find precompiled binaries at https://www.sqlite.org/download.html or install it through your usual package manager.

They might already be installed by default depending on your system as it is often the case with MacOS.

To **install the IDAT python environment** type in a terminal from the root of IDAT package directory:

```
./idat
```

To **uninstall ismpy**, type from the root of IDAT package directory:

```
./idat deinstall
```

When switching to a newer version of IDAT, it is recommended to uninstall and reinstall `ismpy` from the new version (command `./idat deinstall`, followed by `./idat`).

The *ismpy* virtual environment is located at `$HOME/.ismpy/`[2].

Once installed this way, the `python` command that will be used to run IDAT is located at:

```
~/.ismpy/bin/python3
```

Alias for ismpy: It may be useful to define an alias for the ismpy python3 executable. To do so, add in the user `~/.bashrc` or `~/.profile` file:

```
alias ismpy='~/.ismpy/bin/python3'
```

Optional: The list of installed python packages is given below for information, or for users willing to do a manual install:

[2] Actually ismpy is a virtual environment. So the python packages will be installed at the standard location in your computer. The /.ismpy/ directory contains only links with the proper PATHs.

- python (version 3.6 and above)
- h5py (version 2.9 and above)
- pyaml (version 5.1 and above)
- numpy (version 1.16 and above)
- cheroot (version 8.1 and above)
- matplotlib
- sqlite_spellfix

Other requirement: Some IDAT applications require a **modern web browser**. IDAT has been tested with latest versions of Brave, Safari, Firefox and Chrome. Internet Explorer is not supported.

3 The tools in the toolbox

The toolbox includes a graphical user interface (GUI) tool for interactive exploration, extraction and visualization of model results. It includes in addition two command-line tools useful for automating the analysis tasks on large numbers of models: a data extraction tool and a plotting tool.

We recommend that you start by using the interactive tool:

Starting the IDAT Graphical User Interface
You can start using the interactive interface either from the `IDAT` folder with the command

```
./idat dataanalysis
```

or from any folder, using

```
<path_to_IDAT_folder>/idat dataanalysis
```

More generally, the various tools of IDAT are launched through *commands* of the form :

```
./idat <command> [options...]
```

Help on each `<command>` can be obtained with

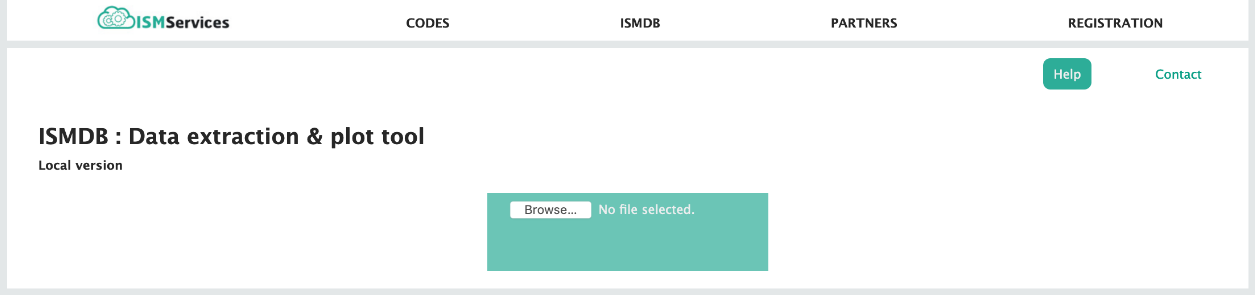
```
./idat <command> -h
```

`<command>`	shortcut	
`dataanalysis`	`da`	**Data Analysis graphical interface** : interactive exploration, visualization and extraction of data from an HDF5 file (Sect. 4)
`plot`	`p`	**Command-line Plot tool** : produces graphs from HDF5 files (Sect. 5)
`extract`	`e`	**Command-line Extractor** : extracts user-queried quantities from an HDF5 file into an ASCII file (Sect. 6)
`deinstall`	`d`	**De-install ismpy** : De-install an existing ismpy python virtual env (Sect. 6)

The use of each of these tools is explained in the corresponding section of the rest of this document.

4 The Data Analysis graphical interface

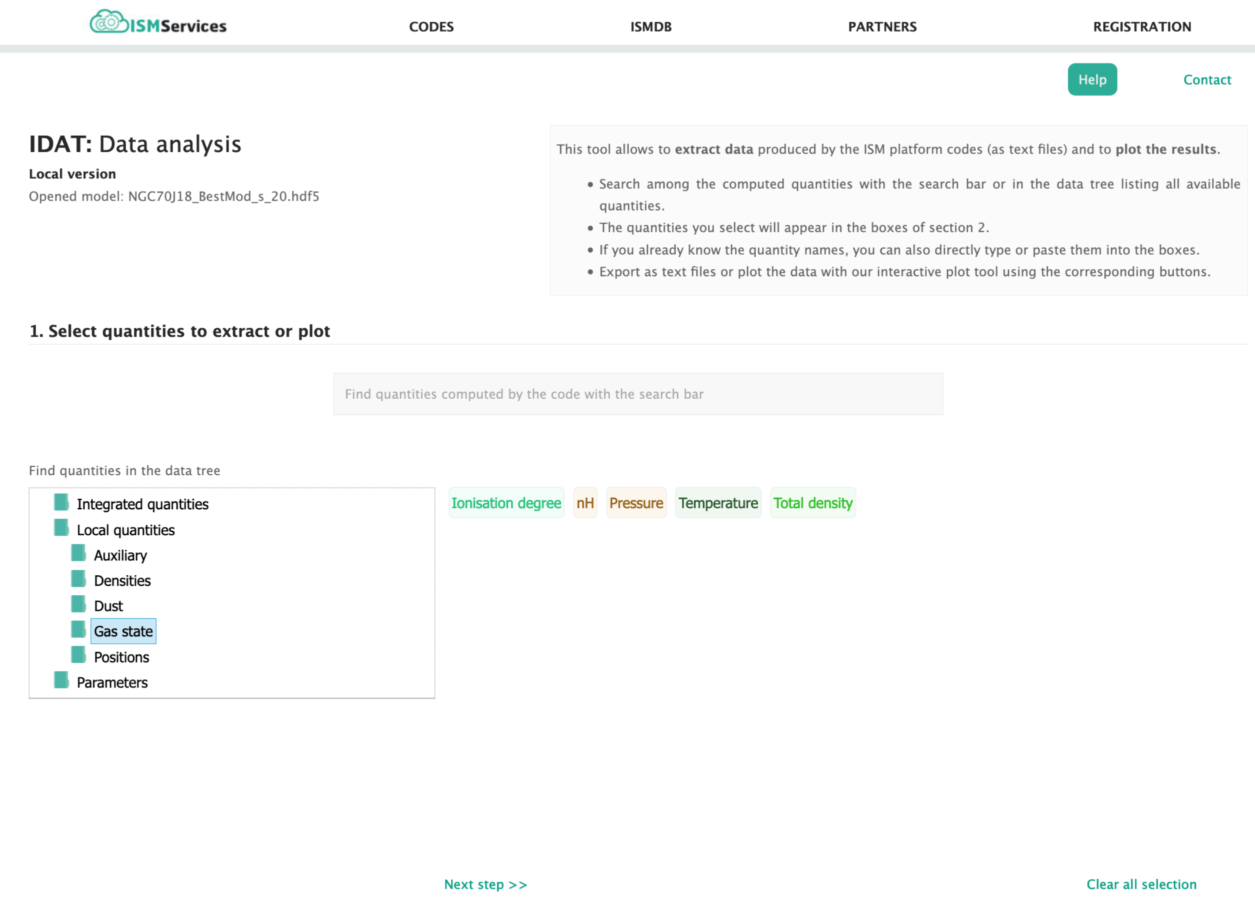
After starting the Graphical Interface (with the command `./idat da` or `./idat dataanalysis`), your default web browser will automatically open and display a page from which you can upload an HDF5 file from anywhere on your disk (shown below).



You can either choose a file from the file browser (by clicking on the "Browse..." button) or use the drag and drop function by dragging a file into the upload area. A new tab will open with an interface to interactively browse, export or plot the data contained in the HDF5 file.

4.1 Select quantities to extract & Plot

After opening a HDF5 file, the data extraction and plot interface appears.



IDAT: Data analysis

Local version

Opened model: NGC70J18_BestMod_s_20.hdf5

This tool allows to **extract data** produced by the ISM platform codes (as text files) and to **plot the results.**

- Search among the computed quantities with the search bar or in the data tree listing all available quantities.
- The quantities you select will appear in the boxes of section 2.
- If you already know the quantity names, you can also directly type or paste them into the boxes.
- Export as text files or plot the data with our interactive plot tool using the corresponding buttons.

1. Select quantities to extract or plot

Find quantities computed by the code with the search bar

Find quantities in the data tree

In the first part of the page (shown above), you can select the quantities you want to extract or plot. You have two ways of searching through the HDF5 file for the quantities you want:

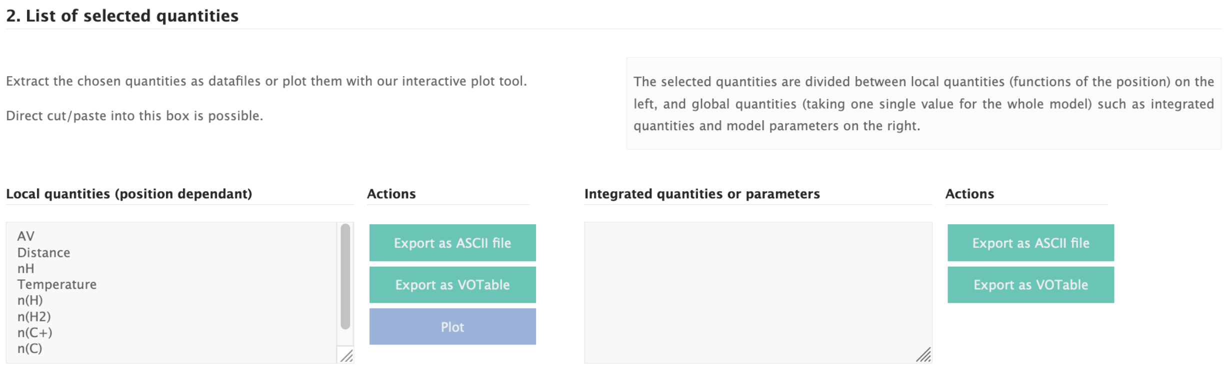
① the **Search Bar** allows you to start typing the name of the quantity you are searching for, and automatic suggestions will appear. Just click on the desired quantity to select it.

② alternatively, the content of the HDF5 file is displayed below the search bar as a **tree**. The left part of the screen shows the folder tree structure. Once reaching one of the bottommost folders, the quantities contained in that folder are displayed on the right, as colored label. Click on the labels of the desired quantities to select them. Click again on already selected labels to deselect them.

The list of quantities that you have selected so far is visible by scrolling down to the second part of the page or by clicking on "Next step »". Clicking "Clear all selection" will deselect all selected quantities.

4.2 Data extraction & Plot

The second part of the interface lists in the left panel the quantities that are spatial profiles (dependent on position in the model) and in the right panel the quantities that are integrated quantities or parameters of the model and that take a single value for the model (not dependent on position, e.g., integrated line intensities).



2. List of selected quantities

Extract the chosen quantities as datafiles or plot them with our interactive plot tool.

Direct cut/paste into this box is possible.

The selected quantities are divided between local quantities (functions of the position) on the left, and global quantities (taking one single value for the whole model) such as integrated quantities and model parameters on the right.

Local quantities (position dependant)

AV
Distance
nH
Temperature
n(H)
n(H2)
n(C+)
n(C)

In both cases, you have the option to extract the chosen quantities as either ASCII or VOTable files by clicking on the corresponding button. It will take a few seconds to generate the file, and a "Download your data file" button will appear once it is ready.

For spatial quantities only, you have the additional possibility to interactively plot and visualize the spatial profiles of the quantities that you have selected. To do so, click the blue "Plot" button. The third part of the page, the plotting interface, will then appear.

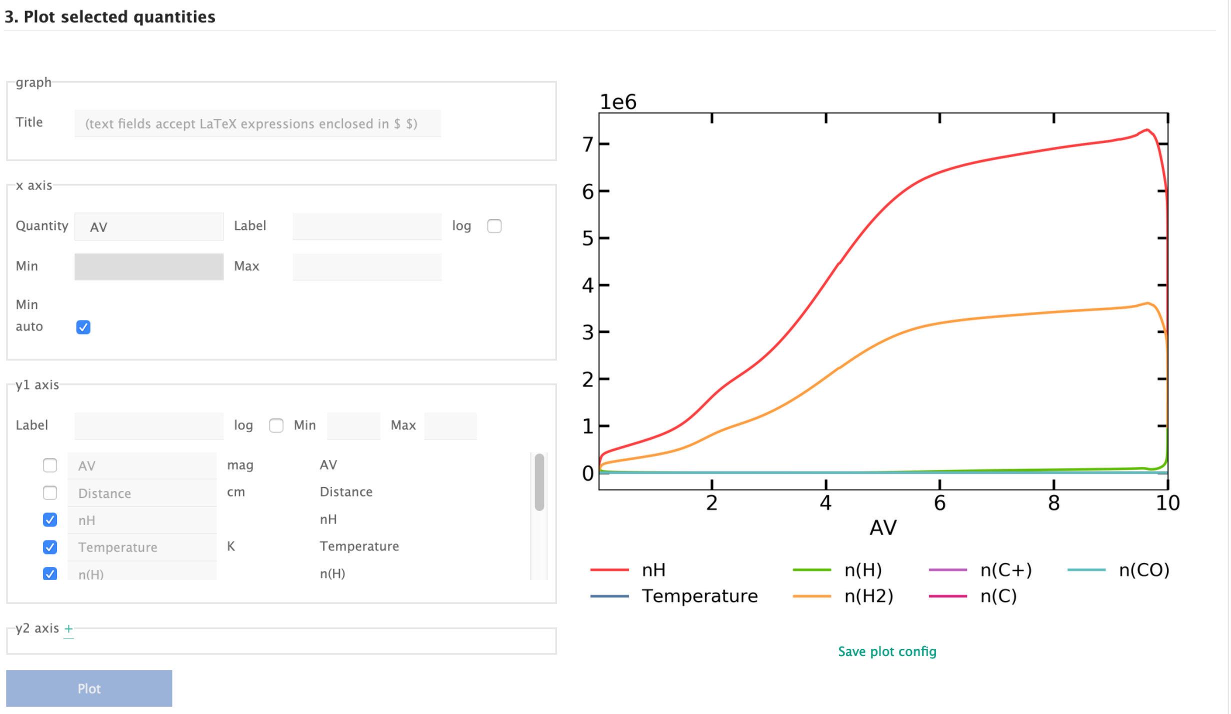
Trick to work more efficiently. To avoid having to go through the search of the same desired quantities again in the future, we recommend to copy and paste the lists obtained in the two boxes of this part of the page into a text file.
Later, while working on another model in the interface, you can then skip part 1 and directly cut and paste these lists from your text file and back into these boxes. Note that you can also edit and delete quantity names directly in these boxes (e.g., you can directly type in a quantity if you know its name, or remove a quantity by deleting the corresponding line).

These ASCII lists of quantities can also be used to call IDAT in command line and extract data without opening the graphical interface.

Spectrum extraction: Due to its specific dimensions, extracting the predicted spectrum (quantities contained in the "Spectrum" subfolder of the HDF5 file) requires a specific procedure.
Select only quantities from the "Spectrum" subfolder (they will appear in the "Integrated quantities or parameters" box) and click one of the two export buttons depending on the desired file format. If you have already selected other quantities, you will need to first remove them from the list before clicking the export button (you can directly edit and delete in the box listing the selected quantities).

4.3 Plot interface

A default plot is produced on the right, that can be customized using the interface on the left. This interface is divided into 3 boxes :



3. Plot selected quantities

graph

Title (text fields accept LaTeX expressions enclosed in $ $)

x axis

Quantity AV Label log □

Min Max

Min auto ☑

y1 axis

Label log □ Min Max

□	AV	mag	AV
□	Distance	cm	Distance
☑	nH		nH
☑	Temperature	K	Temperature
☑	n(H)		n(H)

y2 axis +

① **Graph title.** The first box allows you to set a figure title. LaTeX formulas are allowed, here and in all other text boxes, and must be enclosed in $...$ (you can only use the subset of LaTeX commands accepted by the Matplotlib python modules).

② **Choose X axis quantity.** In the second box, you can choose which quantity you want to use to describe the position (x axis), by clicking on the "Quantity" box. The x-axis label can then be customized in the "Label" box. The x-axis can be set to logarithmic scale by checking the "log" checkbox. X-axis bounds can be set for the graph by entering values in the boxes "Min" and "Max". And finally, the "Min auto" button allows to automatically cut the left part of the profiles where all quantities are constant.

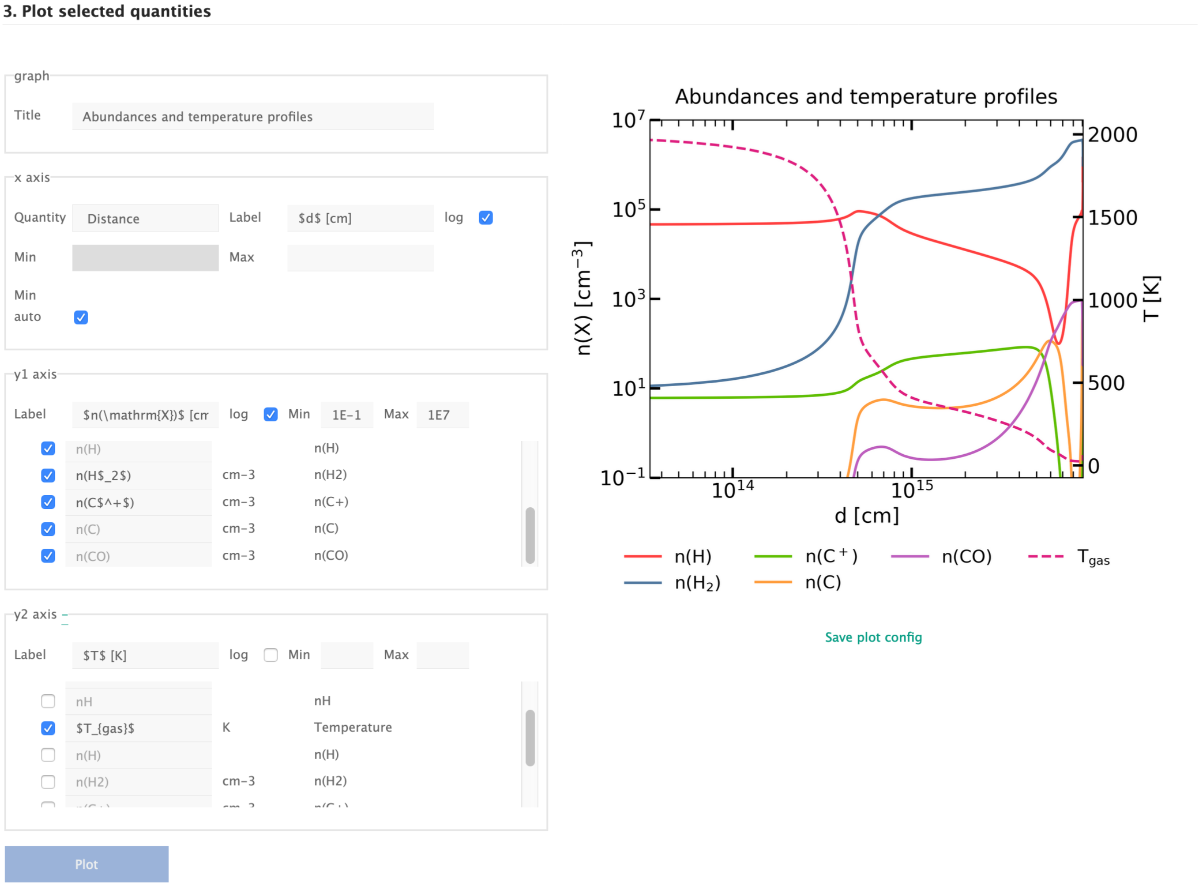
③ **Choose Y axes quantities.** The third part gives you the choice of the plotted quantities:

check the checkboxes in front of the quantities you want to plot among the quantities you have previously selected. As with the x-axis, you can also customize the y-axis label, give bounds, and set the y-axis to logscale. Finaly, you can customize the name that will appear for each quantity in the legend of the figure by typing in the grey area containing the name of the quantity.

Use Latex. Any title and label fields support latex commands. To do so, enclose text in $...$ (you can only use the subset of LaTeX commands accepted by the Matplotlib python modules).

Right Y axis. At the bottom of the interface, clicking on the "+" button in the "y2 axis" box gives you the possibility to set up a second independent y-axis on the right side of the figure, useful for simultaneously visualizing quantities that have different physical units.

After modifying the content of the fields of the interface, click on the "Plot" button at the bottom to visualize the updated figure. Once satisfied with the figure, click on it to download it.



3. Plot selected quantities

graph

Title: Abundances and temperature profiles

x axis

Quantity: Distance — Label: d [cm] — log ☑

Min: — Max:

Min auto ☑

y1 axis

Label: $n(\mathrm{X})$ [cm — log ☑ — Min: 1E−1 — Max: 1E7

☑	n(H)		n(H)
☑	n(H$_2$)	cm−3	n(H2)
☑	n(C$^+$)	cm−3	n(C+)
☑	n(C)	cm−3	n(C)
☑	n(CO)	cm−3	n(CO)

y2 axis −

Label: T [K] — log ☐ — Min: — Max:

☐	nH		nH
☑	T_{gas}	K	Temperature
☐	n(H)		n(H)
☐	n(H2)	cm−3	n(H2)

4.4 Quitting IDAT

To quit IDAT, close the corresponding tab in your browser, return to the terminal you used to start it, and use Ctrl+C to end the process.

5 Extract (command line)

The command-line extractor tool allows to extract selected quantities from the HDF5 file into an ASCII file, in an easily scriptable form for automatizing the treatment of large number of models.

IDAT command line syntax to extract data in ASCII file:

```
./idat extract {.hdf5 file} {quantity selection file} {output file}
```

where you should replace

`{.hdf5 file}` by the path to the HDF5 file from which you want to extract specific quantities.

`{quantity selection file}` by the path to an ASCII file containing the list of quantity names you want to extract, with one name per line.

`{output file}` by the path and name of the output ASCII file to be created.

The quantity names necessary for the quantity selection file can easily be found by first exploring one HDF5 file with the Data Analysis graphical interface (see Sect. 4). The quantity names you have selected can be found in the second part of the page (see Sect. 4.2), and the list of quantities found in the two boxes of this part of the page can be directly copied and pasted "as is" in a text file to produce the necessary quantity selection file. This file can then be used to extract the same selected quantities from a large number of models using the command-line extract tool.

6 Plot (command line)

The Plot tool is a post-processing tool for the codes of the ISM services platform. It produces figures directly from the HDF5 output file, for the spatial profiles of any computed quantity (present in the HDF5 file). As a command-line tool, it can be used to automate the creation of similar graphs for large numbers of models. The quantities to be plotted and the plot format are set through a simple configuration file.

6.1 Running the tool

IDAT command line syntax to plot data:

```
./idat plot {.hdf5 file} {.yaml config file} {output path}
```

where:

- `{.hdf5 file}` is the output HDF5 file produced by one of the codes of the plateform. Note that the Meudon PDR code generates, for each of the last two iterations (or for all iterations if the corresponding flag has been set), two (or three depending on the configuration) HDF5 output files named with the pattern:

  `{model_name}_{'s' or 'c' or 'a'}_{iteration_number}.hdf5.`

The relevant output file for the plot tool is the `_s` output file for the desired iteration. The final results of the code are given by the last iteration. Comparison with previous iterations can help assess whether the computation has fully converged.

- `{.yaml config file}` is the configuration file specifying the quantities to be plotted and the plot format. The syntax of this file is described in the next section. Predefined configuration files are provided in the `modules/plot/conf/` subdirectory inside the `IDAT` directory, which can be used "as is" to get the most usual plots to visualize PDR model results. These predefined config files cover all the main usual visualizations of a PDR model, so that reading the following sections is only necessary if you want to customize the plot presentation, plot unusual quantities, or visualize results from another one of the codes of the plateform.
- `{output path}` should be the path of the directory where for the output figure files should be created (note that ASCII files containing the subset of the data used to create the plot will also be created in the same folder). This argument is optional. By default, the output files are created in the folder where the input HDF5 file is located.

6.2 Syntax of the plot configuration file

Note: manually writing .yaml configuration files can be avoided by using the DataAnalysis graphical interface (see Sect. 4). Open one HDF5 file with the graphical interface and use the plot interface to get the type of graph that you desire. Then, click on the "Save plot config" button beneath the plot to download the corresponding .yaml file. You can then use this configuration file with the command-line Plot tool to produce similar graphs for a large number of models.

The .yaml configuration file is formatted as a list of option name-option value pairs, for instance:

```
plot/y1label: '$ n(\mathrm{H}_2) \quad [\mathrm{cm}^{-3}] $'
```

which contains the option name `plot/y1label` (setting the label for the left Y axis) and its value, here a string with LaTeX syntax.

Minimal options

The most important options are `plot/x` and `plot/y1` which define the quantities to be plotted. The value given must be a string corresponding to the name of a variable in the HDF5 file. The name of all variables contained in the HDF5 files produced by the Meudon PDR code are listed in the `QuantityNames.txt` file.

① **X axis :** `plot/x` requires a single quantity to be given, for instance:

```
plot/x: 'AV'
```

if everything is to be plotted as a function of A_V.

② **Y axes :** `plot/y1` can accept a list of quantities if multiple curves are to be plotted on the same figure. For instance:

```
plot/y1: ['n(H)', 'n(H2)', 'n(C+)', 'n(C)', 'n(CO)', 'n(e-)']
```

will plot 6 curves, corresponding to the densities of H, H_2, C^+, C, CO and e^- across the cloud.

③ **Output file name :**The last main option is `plot/output`, used to choose the name of the output file containing the figure. For instance:

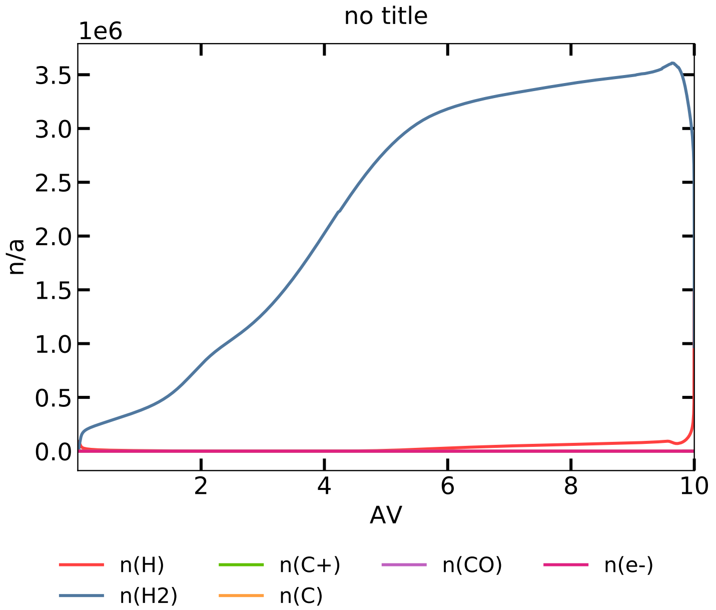
```
plot/output: "plot1"
```

The string given to this option can contain the symbol `MODELNAME` which will be automatically replaced by the model name (contained in the name of the HDF5 file). Two files, a .png and a .pdf version of the same graph, will be generated.

Example of minimal plot configuration file:

```
plot/x:        'AV'
plot/y1:       ['n(H)', 'n(H2)', 'n(C+)', 'n(C)', 'n(CO)', 'n(e-)']
plot/output:   "plot1"
```

The plot tool used with this minimal config file produces a file `plot1.png` and a file `plot1.pdf`, both containing the following figure:



Important formatting options

The first exemple plot shown in the previous subsection lacks important features for readability: axes labels, title, log-scale... The appearance of the plot can be configured through a number of additional options.

- **X label :** `plot/xlabel` is used to set the label of the left X axis. The value must be a string. If this string contains LaTeX syntax, it must be enclosed in single quotes and dollar signs. For instance:

```
plot/xlabel: '$A_V\,[\mathrm{mag.}]$'
```

- **Y label :** `plot/y1label` is used to set the label of the left Y axis (a different right Y axis can be added with the `plot/y2` option described later). The value must be a string. If this string contains LaTeX syntax, it must be enclosed in single quotes and dollar signs. For instance:

  ```
  plot/y1label: '$ n(\mathrm{H}_2) \quad [\mathrm{cm}^{-3}] $'
  ```

- **log scales :** `plot/log:` is used to choose which axes should be displayed in logscale. The value should be a list containing the names of the axis to be log-scaled. Possible names are `'x'` (for the X axis), `'y1'` (for the left Y axis) and `'y2'` (for the secondary right Y axis described later). For instance:

  ```
  plot/log: ['y1']
  ```

  Note that the plot tool automatically replaces the quantities by their logarithm independently of this option if the dynamical range is larger than ten orders of magnitude, in order to avoid overcrowding the axis with log-scale ticks. This automatic log-transformation is disabled if the options `plot/y1min` or `plot/y1max` are set (see below).

- **Plot title :** `plot/title` is used to define a title to the plot (displayed at the top of the figure). The value must be a string. If this string contains LaTeX syntax, it must be enclosed in single quotes and dollar signs. The string given to this option can contain the symbol `MODELNAME` which will be automatically replaced by the model name (contained in the name of the HDF5 file). For instance:

  ```
  plot/title: 'My figure for model MODELNAME'
  ```

- **X min :** `plot/xmin` is used to set the minimum value of the X axis. This allows zooming on a subregion of the plots. Possible values are a float (fixing the minimum of the axis), or the string 'auto', in which case the minimum value is chosen automatically to remove part of the plots where all quantities are mostly constant. For instance:

  ```
  plot/xmin: 1.E-2
  ```

- **X max :**`plot/xmax` is used to set the maximum value of the X axis. This allows zooming on a subregion of the plots. The value must be a float. For instance:

  ```
  plot/xmax: 3.
  ```

- **Y1 min :** `plot/y1min` is used to set the minimum value of the left Y axis. This allows zooming on a subregion of the plots. The value must be a float. For instance:

  ```
  plot/y1min: 1.E-15
  ```

- **Y1 max :** `plot/y1max` is used to set the maximum value of the left Y axis. This allows zooming on a subregion of the plots. The value must be a float. For instance:

  ```
  plot/y1max: 1.E-3
  ```

- **Y data normalization :** `plot/y1normalize` is used to normalize all quantities plotted on the (left) Y axis by a same quantity (also present in the HDF5 file). The value has to be a string containing the name of the quantity by which we wish to normalize. For instance:

  ```
  plot/y1normalize: 'nH'
  ```

Example of a complete plot configuration file:

```
plot/title:        'Chemical profiles'
plot/x:            'AV'
plot/xlabel:       '$A_\mathrm{V}\,[\mathrm{mag.}]$'
plot/y1:           ['n(H)','n(H2)','n(C+)','n(C)','n(CO)','n(e-)']
plot/y1normalize: 'nH'
plot/y1label:      'Relative abundance $n(X)/n_\mathrm{H}$'
plot/xmin:         'auto'
plot/y1min:        1E-8
plot/log:          ['x','y1']
plot/output:       "Relative_densities_MODELNAME"
```

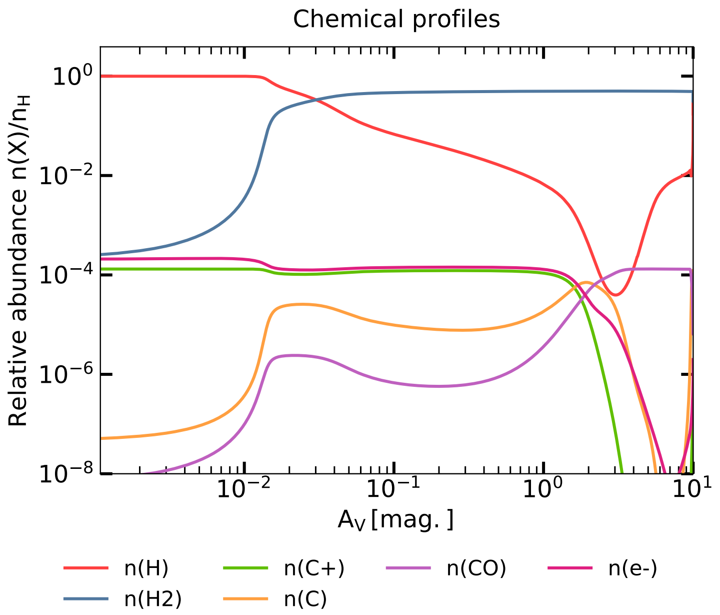
This configuration file would create files with names similar to:

`Relative_densities_PDR152r1809_n_r1e0d1e2A1e0_s_20.png`

and

`Relative_densities_PDR152r1809_n_r1e0d1e2A1e0_s_20.pdf`

(the precise model name will be the one corresponding to the HDF5 file you used), containing the following figure:



Special functionalities

- **Double Y axes:** A second Y axis (display on the right side of the figure) can be added, allowing to plot on the same figure quantities that are in different physical units. This axis uses similar options to the main axis, replacing 'y1' with 'y2': `plot/y2` is used to define the quantities to be plotted with respect to this secondary Y axis, `plot/y2label` to define its label, `plot/y2min` and `plot/y2max` to define its bounds, and `plot/y2normalize` to normalize the quantities related to this axis by a given quantity.

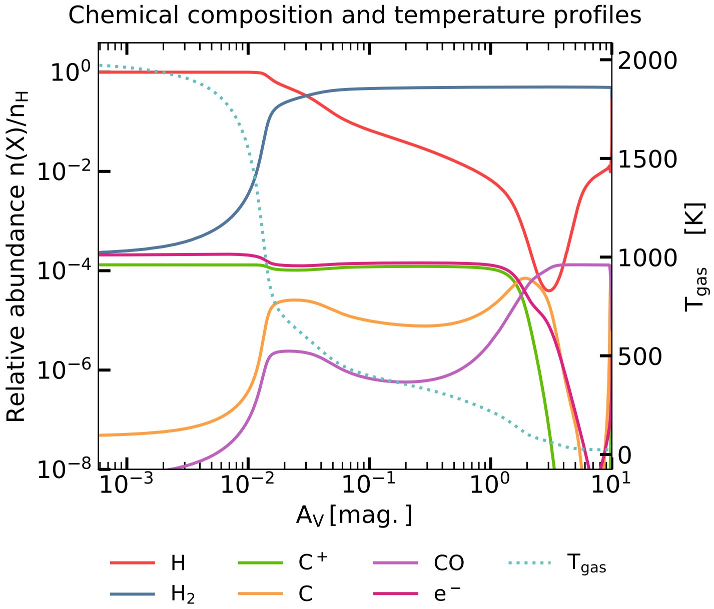
- **Alternate quantity names, curve style and rescaling:** A legend is displayed by default, showing the name of the quantities plotted. The name that are displayed are the names of the quantities in the HDF5 file. These names can be replaced by user-defined names. To do so, replace the name of the quantity (the HDF5 name) in the `plot/y1` option, by a list containing two elements: the HDF5 name and the user-defined name. For instance, replace `’n(H2)’` by `[’n(H2)’,’$\mathrm{H}_2$’]`. This list can be completed by additional informations: a linestyle symbol can be added (`’-’` for solid line, `’--’` for dashed, `’:’` for dotted, `’-.’` for dot-dashed, see matplotlib documentation for more), as well as a float multiplier to be applied to the quantity (e.g. to rescale it to make it comparable with another larger or smaller quantity, or -1 to plot the negative of the quantity). The initial `’n(H2)’` could thus be replaced by `[’n(H2)’,’$\mathrm{H}_2$’,’--’,1.E-4]`. This same functionality applies to `plot/y2` as well.

- **Multiple figures:** Multiple figures can be defined in a single configuration file. The options corresponding to each figure should be grouped in two blocks separated by a line containing only `---`.

Example of a plot configuration file using special functionalities:

```
plot/title:       ’Chemical composition and temperature profiles’
plot/x:           ’AV’
plot/xlabel:      ’$A_\mathrm{V}\,[\mathrm{mag.}]$’
plot/y1:          [
                     [’n(H)’,’H’],
                     [’n(H2)’,’H$_2$’],
                     [’n(C+)’,’C$^+$’],
                     [’n(C)’,’C’],
                     [’n(CO)’,’CO’],
                     [’n(e-)’,’$e^-$’]
                  ]
plot/y1normalize: ’nH’
plot/y1label:     ’Relative abundance $n(X)/n_\mathrm{H}$’
plot/xmin:        ’auto’
plot/y1min:       1E-8
plot/y2:          [ [’Temperatures’,’$T_\mathrm{gas}$’,’:’] ]
plot/y2label:     ’$T_\mathrm{gas}$    [K]’
plot/log:         [’x’,’y1’]
plot/output:      "Relative_densities_MODELNAME"
```

This configuration file would give the following figure:


Chemical composition and temperature profiles
Relative abundance $n(X)/n_H$
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
A_V [mag.]
2000
1500
1000
500
0
T_{gas} [K]
H
H_2
C^+
C
CO
e^-
T_{gas}